

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-151873

(43) Date of publication of application : 30.05.2000

(51)Int.CI. H04N 1/00
B41J 29/38
G06F 3/12

(21)Application number : 10-322658

(71)Applicant : **CANON INC**

(22)Date of filing : 12.11.1998

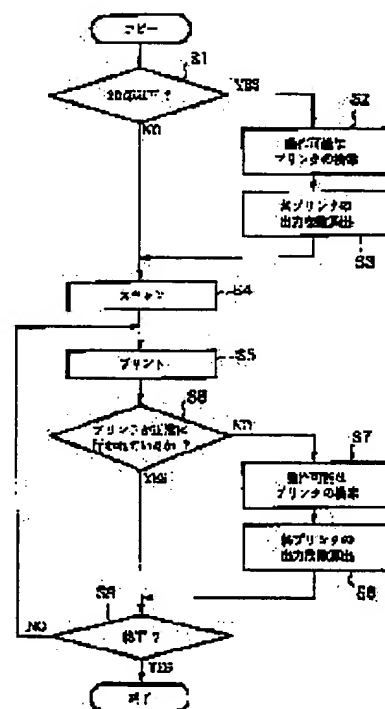
(72)Inventor : SHIMIZU HIDEAKI

(54) INFORMATION PROCESSING SYSTEM AND INFORMATION PROCESSING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide the system where a copy job is conducted in a shortest time by sharing an output job to output destinations that are operative printers depending on their operation capability and deciding the number of distributed jobs in the case that a plurality of printers are connected on a network for copying.

SOLUTION: A scanner and a plurality of printers are interconnected via a LAN, the scanner stores information relating to the connected printers and in the case of sharing an image output to each printer, output possible printers are retrieved based on the stored printer information (S2), the number of shared pages is calculated so that the print-out end time is almost the same in the case of sharing the image output to a plurality of the output possible printers (S3) and a scanned image (S4) is shared to the printers depending on the calculation result to allow the printers to print out the image (S5).



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-151873
(P2000-151873A)

(43) 公開日 平成12年5月30日 (2000.5.30)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
H 0 4 N 1/00		H 0 4 N 1/00	C 2 C 0 6 1
B 4 1 J 29/38		B 4 1 J 29/38	Z 5 B 0 2 1
G 0 6 F 3/12		G 0 6 F 3/12	D 5 C 0 6 2

審査請求 未請求 請求項の数16 O L (全 26 頁)

(21) 出願番号 特願平10-322658

(22) 出願日 平成10年11月12日 (1998.11.12)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 清水 秀昭

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(74) 代理人 100076428

弁理士 大塚 康徳 (外2名)

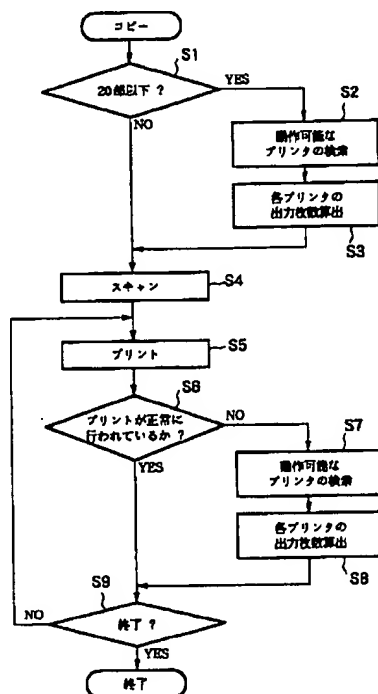
最終頁に続く

(54) 【発明の名称】 情報処理システム及び情報処理方法

(57) 【要約】

【課題】 複数の機器をネットワーク上につなげてコピーを行なう場合に、動作可能なプリンタに対し、その動作能力に応じて出力先を振り分け分配数を決めることにより、コピージョブを最短時間で行うことが出来るシステムを提供する。

【解決手段】 スキャナと複数のプリンタとをLANを介して互いに接続し、スキャナは、接続プリンタに関する情報を保持しておき、各プリンタに画像出力を振り分ける際に、前記保持しているプリンタ情報より出力可能なプリンタを検索し (S2)、可能なプリンタに対して画像を複数のプリンタに振り分けた場合の画像出力終了時間を略同じとなるように振りわけ枚数を算出し (S3)、スキャンした画像 (S4) を算出結果に応じてプリンタに振り分けて画像を出力させる (S5)。



THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号

特開平11-143661

(43) 公開日 平成11年(1999) 5月28日

(51) Int.Cl.⁶

識別記号

F I

G 0 6 F 3/12

G 0 6 F 3/12

C

B 4 1 J 29/38

B 4 1 J 29/38

Z

審査請求 未請求 請求項の数 7 O L (全 30 頁)

(21) 出願番号 特願平9-313178

(22) 出願日 平成9年(1997)11月14日

(71) 出願人 000001443

カシオ計算機株式会社

東京都渋谷区本町1丁目6番2号

(72) 発明者 筒見 勝紀

東京都羽村市栄町3丁目2番1号 カシオ
計算機株式会社羽村技術センター内

(72) 発明者 渡辺 隆保

東京都羽村市栄町3丁目2番1号 カシオ
計算機株式会社羽村技術センター内

(72) 発明者 田村 恒治

東京都羽村市栄町3丁目2番1号 カシオ
計算機株式会社羽村技術センター内

(74) 代理人 弁理士 大菅 義之

最終頁に続く

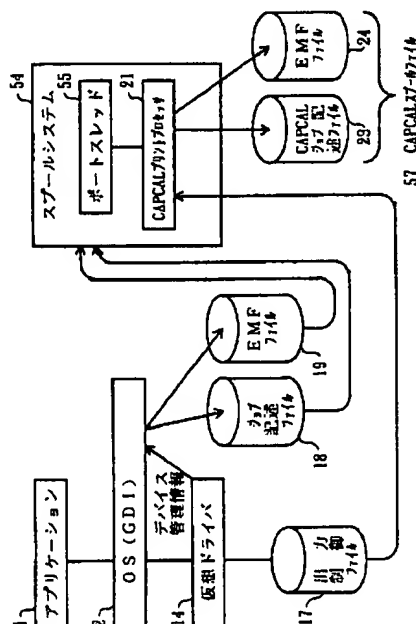
(54) 【発明の名称】 印刷システム

(57) 【要約】

【課題】 汎用アプリケーションで作成した文書をネットワーク上で取り扱うことが可能な印刷システムを提供する。

【解決手段】 仮想ドライバ14を設け、ジョブ記述ファイル18及びEMFファイル19を、GDI2によりスプールさせるとともに、出力制御ファイル17を仮想ドライバ14によりスプールさせることにより、ジョブ記述ファイル18及びEMFファイル19の形式を、Windows標準の形式と一致させ、Windowsのアプリケーションインタフェースを一切変更せずに、汎用パッケージソフトを含むWindows上で動作する全てのアプリケーションについて、ドキュメント配布処理、分散印刷処理、配布先自動検出処理、プリンタエラー及び印刷完了の通知処理等を行う。

仮想ドライバの動作を説明するブロック図



【特許請求の範囲】

【請求項1】 描画データを生成する描画データ生成手段と、

前記描画データ生成手段を管理するオペレーティングシステムと、

前記描画データによるスプールファイルの生成を、前記オペレーティングシステムに対して指示する指示手段と、

前記描画データの印刷条件を設定する印刷条件設定手段と、

前記印刷条件に基づいて、前記描画データの出力制御情報を生成する出力制御情報生成手段と、

前記出力制御情報を格納する出力制御情報格納手段とを備えることを特徴とする印刷システム。

【請求項2】 前記出力制御情報生成手段により生成された出力制御情報に基づいて、前記描画データに対応したプリンタ管理ファイルを生成するプロセッサをさらに備えることを特徴とする請求項1に記載の印刷システム。

【請求項3】 前記プリンタ管理ファイルの内容及び前記スプールファイルの内容を受信する受信手段と、配布先名と出力装置名との対応関係を記述した配布先定義テーブルと、

前記プリンタ管理ファイルの内容から印刷ジョブの検索条件を取得する検索条件取得手段と、

前記検索条件に基づいて、前記印刷ジョブから文字列を抽出する文字列抽出手段と、

前記文字列抽出手段により抽出された文字列と、配布先定義テーブルに記述された配布先名とを比較する比較手段と、

前記配布先名と一致する文字列が抽出された印刷ジョブを、前記配布先名に対応する出力装置に出力する出力手段とを備えることを特徴とする請求項1または2に記載の印刷システム。

【請求項4】 前記印刷条件設定手段は、仮想的な用紙を表示する仮想用紙表示手段と、

前記仮想的な用紙を分割する分割手段と、

前記分割された矩形領域を指定する分割領域指定手段とを備え、

前記文字列抽出手段は、前記分割領域指定手段により指定された矩形領域に対応する印刷ジョブの範囲から、文字列を抽出することを特徴とする請求項3に記載の印刷システム。

【請求項5】 前記プリンタ管理ファイルの内容及び前記スプールファイルの内容を受信する受信手段と、配布先名と出力装置名との対応関係を記述した配布先定義テーブルと、

グループ名と前記配布先名へのリンク情報を記述したグループ定義テーブルと、

前記プリンタ管理ファイルの内容から配布先情報を取得

する配布先情報取得手段と、

前記配布先情報の指定配布モードがグループ指定の場合、前記配布先情報で示されるグループ名とリンクした配布先名を前記グループ定義テーブルから検索し、前記配布先名に対応する出力装置に印刷ジョブを出力する第1の出力手段と、

前記配布先情報の指定配布モードが論理名指定の場合、前記配布先情報で示される論理名と一致する配布先名を前記配布先定義テーブルから検索し、前記配布先名に対応する出力装置に印刷ジョブを出力する第2の出力手段と、

前記配布先情報の指定配布モードが物理名指定の場合、前記配布先情報で示される物理名と一致する出力装置に印刷ジョブを出力する第3の出力手段とを備えることを特徴とする請求項1または2に記載の印刷システム。

【請求項6】 スプールファイルの送信結果を印刷要求元情報とともに記録する送信結果記録手段と、

前記スプールファイルの受信結果を前記印刷要求元情報とともに記録する受信結果記録手段と、

前記スプールファイルの出力結果を前記印刷要求元情報とともに記録する出力結果記録手段と、

前記スプールファイルの印刷結果を前記印刷要求元情報とともに記録する印刷結果記録手段と、

前記スプールファイルの出力先が自己のサーバの管轄下のプリンタである場合、前記印刷結果記録手段に記録された印刷結果を探索し、前記スプールファイルの出力先が他のサーバの管轄下のプリンタである場合、前記送信結果記録手段に記録された送信結果を探索する探索手段とをさらに備えることを特徴とする請求項1～5のいずれか1項に記載の印刷システム。

【請求項7】 描画データを生成する機能と、オペレーティングシステムを実現する機能と、

前記描画データによるスプールファイルを前記オペレーティングシステムに生成させる機能と、

前記描画データの印刷条件を設定する機能と、

前記印刷条件に基づいて、前記描画データの出力制御情報を生成する機能と、

前記出力制御情報を格納する機能とをコンピュータに実行させるプログラムを格納したコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は印刷システムに関し、特に、クライアントサーバ印刷システムに適用して好適なものである。

【0002】

【従来の技術】近年、オフィス業務や基幹業務向けコンピュータのオペレーティングシステムとして、マイクロソフト社のWindowsの普及が目覚ましい。

【0003】図26は、オペレーティングシステムとし

て、Windowsを用いた場合の従来の印刷方法を説明するブロック図である。図26において、ユーザが、アプリケーション201にプリンタドライバ203に対しての印刷を指定した場合、アプリケーション201は、GDI（グラフィカルデバイスインターフェース関数）202に対して印刷命令を実行する。GDI202は、アプリケーション201からの印刷命令が実行されると、プリンタドライバ203から必要な情報を獲得し、スプールファイル（EMFファイルとジョブ記述ファイル）204を作成する。Windowsのスプールシステム205は、指定のポートが印刷可能状態であることを確認すると、デスプール処理を開始する。このデスプール処理により、ポートスレッド206を通じてプリントプロセッサ207に印刷制御が渡る。そして、ローカルプリントプロバイダ208を通じてポートモニタ209に印刷データが渡り、パラレルポートドライバ210を通じて印刷装置211に印刷データが出力される。

【0004】ここで、オペレーティングシステムとして、Windowsを用いた場合、Windowsはパーソナル指向のオペレーティングシステムであることから、基幹業務における大量印刷には不向きな点がある。例えば、1回の印刷アクションで、同時に複数のプリンタに印刷することができない。このため、資料の配布等を行うとき、繰り返し印刷起動をかけ、印刷物を手作業で仕分けし、さらに郵便等で配布する手間がかかる。また、昨今のメールの普及により、ワードプロセッサなどにより作成された少量かつ固定的ドキュメントは、メールで代替えることができるが、アプリケーション稼働による集計データ等の管理帳票の印刷や大量のドキュメントの印刷などは、メールでは代替不可能である。さらに、大量の印刷を行う時、複数のプリンタに分散して印刷することができず、開始頁と終了頁と出力先プリンタとを繰り返し指定する必要がある。

【0005】これらの問題を解決するため、従来では、基幹業務帳票印刷用の専用のAPI（アプリケーション・インターフェース関数）を用意し、そのAPIを用いて、Windowsなどのオペレーティングシステムの有する印刷制御の不備を補う方法があった。

【0006】図27は、従来の分散印刷方法を説明するブロック図である。図27において、アプリケーション221は、専用の印刷制御関数として、API223を呼び出す。ここで、API223には、プリンタ名称、開始頁及び終了頁の組を複数受け取ることができるインターフェースが設られ、API223にこのインターフェースを設けることにより、分散印刷を可能としている。API223が、パラメータ222として、プリンタ名称、開始頁及び終了頁の組を複数受け取るとともに、描画データをアプリケーション221から受け取ると、受け取ったパラメータ222と描画データとを、オ

ペレーティングシステム224及びファイルドライバ225を経由してスプールファイル226に格納する。

【0007】一方、スプールライタ227は、ある決められたディレクトリ（スプールディレクトリ）をポーリングしており、完成したスプールファイル226を検知すると、スプールファイル226の読み出しを開始する。そして、スプールファイル226内に格納されたパラメータ222を取り出すとともに、各ページの描画データをスプールファイル226から順次読み出す。その後、GDI228が呼び出され、パラメータ222の情報に従って、描画データがプリンタドライバ229を介して指定のプリンタ230へ送られる。この結果、指定の頁の印刷を指定のプリンタ230で行うことが可能となり、プリンタ名称、開始頁及び終了頁の組を複数指定しておくことにより、複数のプリンタを用いた分散印刷が可能となる。

【0008】なお、GDI228は、オペレーティングシステム224と密接に結びついたグラフィカルデバイス描画用の関数群である。

【0009】

【発明が解決しようとする課題】しかしながら、従来のAPI223を提供する手法では、API223を呼び出すことが可能な専用のアプリケーション221だけにしか、適用することができなかった。このため、ワードプロセッサなどの汎用パッケージアプリケーションには、API223を呼び出す機能がないため、ワードプロセッサなどの汎用パッケージアプリケーションには適用することができないという問題があった。また、提供されたAPI223を用いて印刷用のアプリケーション221を構築しなければならないという問題もあった。

【0010】さらに、従来の印刷方法では、オペレーティングシステム224の印刷制御の不備からくる問題点があり、API223を提供する方法では、これらの問題点を解決することができなかった。例えば、プリンタが接続されたコンピュータ上の画面には、印刷が完了したことやエラーが発生したことを知らせる通知があるが、通信ネットワークを介して印刷実行を指示したクライアントマシンでは、これらのことが判らない。また、プリンタエラーで停止中のジョブに対し、頁を指定し直した再印刷や出力プリンタを変更した再印刷ができない。さらに、WAN（広域ネットワーク）で接続されたコンピュータ上のプリンタを直接指定して印刷することができない。

【0011】また、オペレーティングシステム224の出力装置・アプリケーションへの依存性に関する問題点として、以下のものがあった。まず、プリンタ機種やプリンタメーカを特定して印刷機能が提供されており、ネットワーク環境の多様なプリンタを有効に使うためには、多大な手間がかかった。このため、特定出力装置（プリンタなど）のスペックに基づきアプリケーション

により作成したドキュメントは、出力装置の物理仕様に依存したものとなり、可搬性がなかった。この結果、物理仕様の異なる出力装置に出力すると、期待する結果を得られない場合があった。

【0012】次に、システム上の全アプリケーションで出力したドキュメントを対象とした場合、アプリケーションで出力後は、何もすることができなかった。例えば、出力先の追加・変更、頁単位での操作、出力順の操作など様々なサービスが不可能であった。

【0013】また、オペレーティングシステム224の配布印刷方式に関する問題点として、以下のものがあった。まず、印刷出力（ドキュメント）を配布する場合、配布先のプリンタを直接指示するものであり、システム環境（配布先に対するプリンタ名称）を常に意識する必要があった。また、配布対象となるプリンタの名称を全て指定したり、ドキュメント毎に配布先を指定したりすることは、ユーザにとって負担になっていた。

【0014】次に、ドキュメント内容から配布先を決定する場合、配布先を決定するためのドキュメントの領域を指定する必要があるが、指定した領域の文字列全てが配布先候補として扱われるので、確実に配布先を指示するためには、指定した領域で利用できる文字列に制限が発生する。このため、ある配布先を一時的に対象外とする場合、ドキュメントから文字列を削除するなどの複雑な操作や後の修復の手間が必要となり、誤操作の要因となる場合があった。さらに、自由度が高いため実現方法が複雑で性能などに課題が発生する。

【0015】また、オペレーティングシステム224の領域指定方式に関する問題点として、以下のものがあった。何からの目的で印刷ドキュメントの領域を指定することが必要な場合、各アプリケーションにその機能を組み込む必要があるが、各ドキュメントの形状（用紙サイズ、置き方）に依存した設定方式であるため、ドキュメント毎に設定を行う必要があった。さらに、一度設定しても、ドキュメントの形状を変更するたびに再設定するという手間があった。

【0016】また、オペレーティングシステム224のドキュメント出力監視制御方式に関する問題点として、以下のものがあった。印刷処理（プリンタ出力）の管理であり、既存のシステムでは、プリンタを主体とした監視機能は提供されていたが、各ユーザが投入した印刷処理を主体とした監視が不可能であり、ユーザは、自分が投入した印刷処理全てを、簡単に監視することが困難であった。

【0017】そこで、本発明の第1の目的は、汎用アプリケーションで作成した文書をネットワーク上で取り扱うことが可能な印刷システムを提供することである。また、本発明の第2の目的は、汎用アプリケーションで作成した文書の配布印刷を容易に行うことが可能な印刷システムを提供することである。

【0018】また、本発明の第3の目的は、ネットワークに接続されているプリンタの動作状態をクライアント側で監視することが可能な印刷システムを提供することである。

【0019】

【課題を解決するための手段】上述した課題を解決するために、本発明によれば、描画データを生成する描画データ生成手段と、前記描画データ生成手段を管理するオペレーティングシステムと、前記描画データに基づくスプールファイルの生成を、前記オペレーティングシステムに対して指示する指示手段と、前記描画データの印刷条件を設定する印刷条件設定手段と、前記印刷条件に基づいて、前記描画データの出力制御情報を生成する出力制御情報生成手段と、前記出力制御情報を格納する出力制御情報格納手段とを備えている。

【0020】このことにより、描画データ生成手段により生成された描画データを、その描画データ生成手段を管理するオペレーティングシステムによりスプールさせることが可能となることから、そのオペレーティングシステム上で動作するアプリケーションならば、どのようなアプリケーションも、描画データ生成手段として使用することが可能となる。また、出力制御情報を格納する出力制御情報格納手段を設けることにより、出力制御情報を描画データに基づくスプールファイルと別々に格納することが可能となることから、オペレーティングシステムに備わっていなかった出力制御機能を、後から自由に追加することが可能となる。この結果、従来のオペレーティングシステムでは容易に実現することができなかったネットワーク環境でのドキュメント配布処理、分散印刷処理、配布先自動検出処理、またはプリンタエラー及び印刷完了の通知処理等を、汎用アプリケーションで作成した文書に対して行うことが可能となる。

【0021】また、本発明の一態様によれば、前記出力制御情報生成手段により生成された出力制御情報に基づいて、前記描画データに対応したプリンタ管理ファイルを生成するプロセッサをさらに備えている。

【0022】このことにより、オペレーティングシステムと独立した独自の形式でプリンタ管理ファイルの処理や操作を行うことが可能となり、ドキュメントの印刷処理を多種多様な形態で柔軟に行うことが可能となる。

【0023】また、本発明の一態様によれば、配布先名と出力装置名との対応関係を記述した配布先定義テーブルと、受信したプリンタ管理ファイルの内容から印刷ジョブの検索条件を取得する検索条件取得手段と、前記検索条件に基づいて、前記印刷ジョブから文字列を抽出する文字列抽出手段と、前記文字列抽出手段により抽出された文字列と、配布先定義テーブルに記述された配布先名とを比較する比較手段と、前記配布先名と一致する文字列が抽出された印刷ジョブを、前記配布先名に対応する出力装置に出力する出力手段とを備えている。

【0024】このことにより、クライアントから送られた検索条件に基づいて、ドキュメントに含まれる予め指定された文字列を自動的に判別することが可能となり、印刷ジョブから配布先を自動的に検出することが可能となるとともに、配布先定義テーブルを参照することにより、自動的に検出された配布先に基づいて、出力装置を特定することが可能となる。この結果、印刷ドキュメントの配布先を指定する手間を省くことが可能となり、配布先に対応するプリンタ名称を意識することなく、1回の印刷指示を行うだけで検索条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0025】また、本発明の一態様によれば、仮想的な用紙を表示する仮想用紙表示手段と、前記仮想的な用紙を分割する分割手段と、前記分割された矩形領域を指定する分割領域指定手段とを備え、前記分割領域指定手段により指定された矩形領域に対応する印刷ジョブの範囲から、文字列を抽出するようにしている。

【0026】このことにより、検索領域を用紙の上下左右や真ん中などの方向で指定することが可能となることから、配布先の検索などのため、印刷ドキュメントの領域を指定することが必要な場合、仮想的な用紙への1回の領域設定により、各印刷ドキュメントの形状（用紙サイズ、置き方）に依存することなく、配布先の検索領域を適切に設定することが可能となる。

【0027】また、本発明の一態様によれば、配布先名と出力装置名との対応関係を記述した配布先定義テーブルと、グループ名と前記配布先名へのリンク情報を記述したグループ定義テーブルと、受信したプリンタ管理ファイルの内容から配布先情報を取得する配布先情報取得手段と、前記配布先情報の指定配布モードがグループ指定の場合、前記配布先情報で示されるグループ名とリンクした配布先名を前記グループ定義テーブルから検索し、前記配布先情報の指定配布モードが論理名指定の場合、前記配布先情報で示される論理名と一致する配布先名を前記配布先定義テーブルから検索し、前記配布先情報の指定配布モードが物理名指定の場合、前記配布先情報で示される物理名と一致する出力装置に印刷ジョブを出力するようにしている。

【0028】このことにより、N台分の配布先名と出力装置との対応関係を登録したり、1つの論理的な配布先に複数の出力装置を定義することが可能となり、配布先を1つ指定するだけで、その配布先に対応する複数の出力装置を自動的に決定することが可能となることから、配布先を物理的な出力装置ではなく、論理的な名称で指定することが可能となるとともに、1回の指定により複数の出力装置にドキュメントを配布することが可能となる。このため、希望する配布先を印刷実行時に複数選択したり、印刷ドキュメント毎に配布先を指定したり、配布対象であるプリンタの名称を全て指定したりする手間を省くことが可能となり、配布先に対応するプリンタ名

称を意識することなく、1回の印刷指示で条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0029】また、本発明の一態様によれば、スプールファイルの送信結果を印刷要求元情報とともに記録する送信結果記録手段と、前記スプールファイルの受信結果を前記印刷要求元情報とともに記録する受信結果記録手段と、前記スプールファイルの出力結果を前記印刷要求元情報とともに記録する出力結果記録手段と、前記スプールファイルの印刷結果を前記印刷要求元情報とともに記録する印刷結果記録手段と、前記スプールファイルの出力先が自己のサーバの管轄下のプリンタである場合、前記印刷結果記録手段に記録された印刷結果を探索し、前記スプールファイルの出力先が他のサーバの管轄下のプリンタである場合、前記送信結果記録手段に記録された送信結果を探索する探索手段とを備えているこのことにより、ネットワーク上でのスプールファイルの転送が行われた場合においても、そのスプールファイルの送信結果、受信結果、出力結果及び印刷結果が、どの印刷要求元に起因して得られたのかを容易に判定することが可能となり、送信結果記録手段、受信結果記録手段、出力結果記録手段及び印刷結果記録手段の内容を調べることにより、自分の印刷要求がどのサーバのプリンタでどのように処理されたのかを認識することが可能となる。この結果、クライアントは、プリンタがどのサーバで管理されているのかを意識することなく、印刷要求を行うことが可能となるとともに、いずれのサーバにおいて印刷が行われた場合においても、印刷状況を正確に把握することが可能となる。

【0030】

【発明の実施の形態】以下、本発明の一実施例に係わる印刷システムについて図面を参照しながら説明する。なお、以下の実施例では、Windowsによるオペレーティングシステム下における出力装置運用支援（以下CAPCALと称する）制御方式について説明する。また、このCAPCAL制御方式では、システム上の全てのアプリケーションの出力ドキュメントを処理対象とすることができる。

【0031】図1は、本発明の一実施例に係わる印刷システムの構成を示すブロック図である。図1において、クライアント300とサーバ301とが通信ネットワーク302、303を介して接続され、サーバ301はプリンタ9と接続されている。ここで、通信ネットワーク302は、例えば、LAN（ローカルエリアネットワーク）、通信ネットワーク303は、例えば、LAN、WAN（ワイドエリアネットワーク）、インターネットである。クライアント300には、アプリケーション1、GDI2、Windows印刷制御部3及びクライアント制御部4が設けられ、サーバ301には、Windows印刷制御部5、サーバ制御部6、GDI7及びプリ

ンタ出力制御部8が設けられている。そして、Windowsによる印刷処理を行う場合、Windows印刷制御部3、5による処理が行われ、CAPCAL制御を行う場合、クライアント制御部4、サーバ制御部6及びプリンタ出力制御部8による処理が行われる。なお、CAPCAL制御を選択する場合、CAPCAL制御用に設けられたCAPCALドライバをアプリケーション1で指定してGDI2を呼び出す。

【0032】図2は、図1のクライアント制御部4の構成を説明するブロック図である。図2において、クライアント制御部4には、CAPCALドライバ14、CAPCALドライバプロパティ制御スレッド15、出力制御ファイル17、ジョブ記述ファイル18、EMFファイル19、24、CAPCALクライアント定義ファイル20、CAPCALプリントプロセッサ21、CAPCALジョブ記述ファイル23、CAPCAL転送先定義ファイル25、CAPCALクライアントメインプロセス26及びCAPCALジョブ転送スレッド27が設けられている。また、Windows印刷制御部3には、スーパープロセス11、ローカルプリントプロバイダ12及びプリントプロセッサ13が設けられている。ここで、クライアント制御部4は、Windowsによるオペレーティングシステム上で稼働するアプリケーション1が描画した印刷データをCAPCALの処理環境に取り込み、サーバ301へ転送するまでの処理を制御する。

【0033】アプリケーション1が描画した印刷データをCAPCAL環境で処理する場合、アプリケーション1は、仮想ドライバとしてのCAPCALドライバ14を指定し、GDI2を呼び出す。GDI2は、CAPCALドライバ14から描画機能を取得し、アプリケーション1の要求した描画プリミティブをCAPCALドライバ14が有する描画機能に変換して、システムスプールキュー16にスプールする。このシステムスプールキュー16には、印刷要求したユーザ名や、アプリケーション名などを格納したジョブ記述ファイル18と、描画情報をフォーマット化して格納したEMFファイル19が作成される。EMFとは、Enhanced Meta-Fileの略で、Windowsにおける描画データの論理的な格納形式である。なお、ジョブ記述ファイル18及びEMFファイル19の形式は、Windows標準の形式と完全に同一である。このため、ジョブ記述ファイル18及びEMFファイル19は、Windowsに備わっているデフォルト機能により生成させることができる。

【0034】また、CAPCALドライバ14は、CAPCAL特有の出力制御情報を出力制御ファイル17として、システムスプールキュー16に書き込む。なお、この出力制御情報は、CAPCALドライバ14側からダイアログ画面を開き、配布や分散などの印刷条件をオ

ペレータに指定させることにより、設定することができる。ここで、オペレータにより入力された出力制御情報を文書ごとに記憶し、次回にその文書が印刷される時は、前回入力されたその文書の出力制御情報をデフォルトとして使用するようにしても良い。

【0035】アプリケーション1の全描画データのスプールが完了すると、CAPCALプリントプロセッサ21が呼び出される。このCAPCALプリントプロセッサ21の呼び出しは、アプリケーション1でCAPCALドライバ14を指定することにより行われる。なお、アプリケーション1で他のドライバが指定された場合は、Windowsに備わっているプリントプロセッサ13が呼び出される。

【0036】CAPCALプリントプロセッサ21は、システムスプールキュー16からジョブ記述ファイル18と出力制御ファイル17の内容を取り込み、CAPCALスプールキュー22にCAPCALジョブ記述ファイル23として書き込む。ここで、CAPCALクライアント定義ファイル20には、出力制御情報のデフォルト値が定義されており、出力制御ファイル17に出力制御情報が格納されていない場合は、CAPCALクライアント定義ファイル20から出力制御情報を取り込む。

【0037】また、CAPCALプリントプロセッサ21は、システムスプールキュー16のEMFファイル19を、そのままCAPCALスプールキュー22のEMFファイル24に書き込む。CAPCALジョブ記述ファイル23及びEMFファイル24が、CAPCALスプールキュー22へ書き込まれると、CAPCALプリントプロセッサ21は、このことをCAPCALジョブ転送スレッド27に通知する。

【0038】CAPCALジョブ転送スレッド27は、予め定義された転送手段に基づいて、CAPCALジョブ記述ファイル23及びEMFファイル24をサーバ301に転送する。なお、転送手段及び転送先のサーバ名は、CAPCAL転送先定義ファイル25に格納されており、CAPCALジョブ転送スレッド27は、CAPCAL転送先定義ファイル25の内容に従って、転送手段及び転送先のサーバを決定する。なお、転送手段として、ファイル転送プロトコル(ftp)を用いた通信ネットワーク28やメールの添付ファイルを用いた通信ネットワーク29が使用可能である。また、これら以外の転送手段を用いても良い。

【0039】このように、仮想ドライバとしてのCAPCALドライバ14をクライアント制御部4に設ける。そして、ジョブ記述ファイル18及びEMFファイル19を、システムスプールキュー16にGDI2によりスプールさせるとともに、出力制御ファイル17をCAPCALドライバ14が独自にシステムスプールキュー16にスプールする。このことにより、ジョブ記述ファイル18及びEMFファイル19の形式を、Window

s標準の形式と一致させることが可能となり、Windowsのアプリケーションインタフェースを一切変更せずに、汎用パッケージソフトを含むWindows上で動作する全てのアプリケーションについて、ドキュメント配布処理、分散印刷処理、配布先自動検出処理、プリンタエラー及び印刷完了の通知処理等を実現することが可能となる。この結果、ドキュメント配布処理、分散印刷処理、配布先自動検出処理、プリンタエラー及び印刷完了の通知処理等を実現するために、印刷用のアプリケーションを新たに構築する必要がなくなる。

【0040】図3は、図1のサーバ制御部の構成を説明するブロック図である。図3において、サーバ制御部6には、メール用のCAPCAL受信スレッド30、ftp用のCAPCAL受信スレッド31、CAPCALサーバメインプロセス32、CAPCALジョブ記述ファイル34、39、EMFファイル35、40、CAPCALサーバ側定義ファイル36、配布分散処理スレッド37、プリンタ別出力スレッド41、42及びCAPCALジョブ転送スレッド43が設けられている。

【0041】CAPCALサーバメインプロセス32によりCAPCAL受信スレッド30、31が起動されると、CAPCAL受信スレッド30、31は、CAPCALジョブ記述ファイル23及びEMFファイル24をCAPCALジョブ転送スレッド27から受信する。そして、受信したCAPCALジョブ記述ファイル23及びEMFファイル24の内容を、CAPCALスプールドキュメントキュー33のCAPCALジョブ記述ファイル34及びEMFファイル35に書き込む。ここで、CAPCAL受信スレッド30、31は、ファイル名（ジョブID）が重複しないように管理を行う。CAPCALスプールドキュメントキュー33への書き込みが完了したら、CAPCALサーバメインプロセス32は配布分散処理スレッド37を起動し、CAPCALスプールドキュメントキュー33への書き込み完了を、配布分散処理スレッド37に通知する。

【0042】配布分散処理スレッド37は、CAPCALジョブ記述ファイル34から出力制御情報を取り込む。また、CAPCALサーバ側定義ファイル36の内容を予め読み込んでおく。そして、配布分散処理スレッド37は、出力制御情報及びサーバ側定義ファイルの内容から、出力プリンタ及び出力する開始頁、終了頁を決定し、プリンタ9ごとに設けられたCAPCALスプールプリントキュー38に印刷要求を投入する。CAPCALサーバ側定義ファイル36には、配布先名とプリンタ名（またはサーバ名）との対応関係が格納されており、配布先が論理名で特定されている場合においても、CAPCALサーバ側定義ファイル36を参照することにより、出力プリンタを決定することができる。CAPCALスプールプリントキュー38へのスプールが完了したら、印刷要求があったプリンタ9を専門に受け持つ

プリンタ別出力スレッド41に通知する。

【0043】プリンタ別出力スレッド41は、CAPCALスプールプリントキュー38からEMFファイル40を取り出し、GDI7を呼び出すことにより、Windows印刷制御部5に描画データを送る。なお、出力制御情報及びCAPCALサーバ側定義ファイル36の内容により、出力プリンタとして、サーバ301が管理しているプリンタ9以外のプリンタが指定されている場合、配布分散処理スレッド37は、CAPCALジョブ転送スレッド43から通信ネットワーク44を介し、指定されているプリンタを管理しているサーバに印刷要求を送る。このように、CAPCALサーバ側定義ファイル36に、配布先名とプリンタ名（またはサーバ名）との対応関係を格納しておくことにより、配布先が論理名で特定されている場合においても、出力プリンタを決定することが可能となり、ユーザは、出力プリンタ名を意識することなく、配布印刷を行うことが可能となる。

【0044】また、プリンタ別出力スレッド41、42を各プリンタごとに設け、プリンタ別出力スレッド41、42がエミュレーションを行うことにより、プリンタの仕様が異なる場合においても、適切な出力結果を得ることが可能となり、プリンタ機種やプリンタメーカーを意識することなく、ネットワーク上の様々のプリンタを利用して印刷を行うことが可能となる。

【0045】図4は、図1のプリンタ出力制御部の構成を説明するブロック図である。図4において、プリンタ出力制御部8には、セントロポートモニタ48及びプリンタ出力管理部50が設けられ、サーバ制御部6には、CAPCALサーバメインプロセス32、EMFファイル40及びプリンタ別出力スレッド41が設けられ、Windows印刷制御部5には、Rawスプールファイル46、プリントプロセッサ47及びスプーラプロセスメイン49が設けられ、クライアント300には、CAPCALクライアントメインプロセス26が設けられている。

【0046】プリンタ別出力スレッド41は、CAPCALスプールプリントキュー38からEMFファイル40を取り出し、GDI7を呼び出すことにより、Windows印刷制御部5に描画データを送る。GDI7は、印刷指定されたプリンタ9用のプリンタドライバ45と交信し、印刷指定されたプリンタ9専用のコマンド（ESCシーケンス）にEMFファイル40を変換する。変換されたESCシーケンスは、Windowsのシステムスプールキューに格納される。これをRawスプールファイル46と言う。Rawスプールファイル46は、純粋なプリンタコマンド（ESCシーケンス）だけで構成される。Rawスプールファイル46は、Windowsのスプーラプロセスメイン49によりプリントプロセッサ47で読み込まれ、1頁分の描画データ単位で、セントロポートモニタ48に送られる。

【0047】プリンタ9とサーバ302とのインタフェースがLANの場合、LANポートモニタに送られ、その他のインタフェースの場合、そのインタフェース用のポートモニタが送られる。なお、ここまでの処理は、Windows印刷制御部5によるデータ処理と同一である。

【0048】セントロポートモニタ48は、スプールプロセスメイン49経由で受け取ったプリンタ9のコマンドデータを、セントロポートに出力する。ここで、プリンタ9にエラーが発生し、コマンドデータを出力できなかった場合、セントロポートモニタ48は、プリンタ出力管理部50に、プリンタ名称、エラーコード及び元のドキュメントを識別するジョブIDを通知する。また、セントロポートモニタ51も、そのセントロポートモニタ51に接続されているプリンタにエラーが発生した場合、プリンタ出力管理部50に、プリンタ名称、エラーコード及び元のドキュメントを識別するジョブIDを通知する。ここで、ジョブIDは、Windows印刷制御の仕組みに存在するプリントジョブを特定するIDであり、ジョブIDをプリンタ出力管理部50に通知することにより、プリンタ出力管理部50は、元のドキュメント名称、印刷要求したクライアントホスト名称及びユーザ名称を取得することができる。

【0049】プリンタ出力管理部50は、プリンタ名称、エラーコード及びジョブIDをセントロポートモニタ48から受け取ると、CAPCALサーバメインプロセス32に、これらの情報を通知する。

【0050】CAPCALサーバメインプロセス32は、取得したジョブIDから印刷要求を発行したクライアントホスト名を検出し、そのクライアントホスト上で稼働するCAPCALクライアントメインプロセス26にドキュメント名称、ユーザ名及びエラーコードを通知する。また、画面呼び出し52を行うことにより、これらの情報を表示する。

【0051】CAPCALクライアントメインプロセス26は、CAPCALサーバメインプロセス32からドキュメント名称、ユーザ名及びエラーコードを受信すると、画面呼び出し53を行うことにより、ユーザ名、印刷中のドキュメント名称、エラーメッセージを表示する。

【0052】また、セントロポートモニタ48は、1つのジョブの出力が完全に終了したら、プリンタ出力管理部50にプリンタ名称及びジョブIDを通知する。プリンタ出力管理部50は、セントロポートモニタ48からプリンタ名称及びジョブIDを受け取ると、CAPCALサーバメインプロセス32に、これらの情報を通知する。

【0053】CAPCALサーバメインプロセス32は、取得したジョブIDから印刷要求を発行したクライアントホスト名を検出し、そのクライアントホスト上で

稼働するCAPCALクライアントメインプロセス26に、ドキュメント名称及びユーザ名を通知する。また、画面呼び出し52を行うことにより、これらの情報を表示する。

【0054】CAPCALクライアントメインプロセス26は、CAPCALサーバメインプロセス32からドキュメント名称及びユーザ名を受信すると、画面呼び出し53を行うことにより、ドキュメントの印刷が完了したことを示すメッセージを表示する。

【0055】このように、セントロポートモニタ48は、元のドキュメントを識別するジョブIDをプリンタ出力管理部50に通知することにより、プリンタ出力管理部50は、印刷要求したクライアントホスト名を認識することが可能となり、印刷要求したクライアント300に対し、印刷エラーや印刷完了を通知することが可能となることから、ドキュメントの印刷状況をクライアント300側で認識することが可能となる。

【0056】以下、図1の印刷システムについて、より詳細に説明する。図5は、仮想ドライバ14の動作を説明するブロック図であり、CAPCALスプールファイル57が作成されるまでの処理の流れを示す。

【0057】図5において、ユーザが、アプリケーション1に仮想ドライバ14に対しての印刷を指定した場合、アプリケーション1は、GDI2に対して印刷命令を実行する（CAPCALの場合は自動的にスプール印刷となる）。GDI2は、印刷命令が実行されると、デバイス管理情報（Windowsシステム概値。プリンタドライバ独自の情報を付加可能。以下、DEVMODE情報と記述する）を仮想ドライバ14に要求する。仮想ドライバ14は、メーカーやプリンタ機種に依存しない共通な情報（用紙サイズ、用紙種等）と、出力制御ファイル17の名称をDEVMODE情報に設定し、GDI2に返す。GDI2は、仮想ドライバ14から必要な情報を受け取ると、DEVMODE情報と描画情報（GDI INFO情報：プリンタドライバ描画能力。プリンタドライバがGDI2に返す情報。Windowsシステム概値）によって、EMFファイル19とジョブ記述ファイル18を作成する。このように、CAPCAL制御印刷では、GDI2が、DEVMODE情報を仮想ドライバ14から受け取ることにより、Windowsシステムが、ジョブ記述ファイル（Windowsシステム概値）18を作成する。

【0058】一方、仮想ドライバ14は、仮想ドライバ14のプリンタプロパティ（指定配布、自動配布、分散印刷の情報を指定）や、予め保存してある出力制御情報により、DEVMODE情報に付加されたファイル名称で出力制御ファイル17を作成する。Windowsのスプールシステム54は、指定されたポートが印刷可能状態であることを確認して、デスプール処理を開始する。デスプール処理により、ポートスレッド55を通じ

てCAPCALプリントプロセッサ21に制御が渡る。CAPCALプリントプロセッサ21は、Windowsシステムが作成したジョブ記述ファイル18を読み込むことにより、DEVMODE情報、仮想ドライバ3によりDEVMODE情報に付加された出力制御ファイル17の名称、及びEMFファイル19の名称を取得する。取得した出力制御ファイル17の名称及びEMFファイル19の名称に基づいて、ジョブ記述ファイル18及びEMFファイル19から出力制御情報と印刷データをそれぞれ取得し、CAPCALスプールファイル57(CAPCALジョブ記述ファイル23及びEMFファイル24)を作成する。

【0059】CAPCALプリントプロセッサ7は、CAPCALスプールファイル57を作成すると、CAPCALスプールファイル57の内容をCAPCALジョブ転送スレッド27を介してサーバ301に送ることにより、指定配布印刷、自動配布印刷、分散印刷等の機能を実現する。

【0060】このように、仮想ドライバ14は、設定されたDEVMODE情報をGDI2に送り、GDI2によりジョブ記述ファイル18とEMFファイル19とを生成させるとともに、出力制御情報を出力制御ファイル17に独自に格納し、独自に格納した出力制御情報をCAPCALプリントプロセッサ21に渡すようにしている。

【0061】この結果、アプリケーション1により生成された描画データを、オペレーティングシステム2によりスプールさせることが可能となることから、そのオペレーティングシステム2上で動作するアプリケーション1ならば、どのようなアプリケーション1も使用することが可能となるとともに、ジョブ記述ファイル18及びEMFファイル19とは別個に出力制御情報を格納することにより、オペレーティングシステム2に備わっていない出力制御機能を、後から自由に追加することが可能となり、ネットワーク環境でのドキュメント配布処理、分散印刷処理、配布先自動検出処理、またはプリンタエラー及び印刷完了の通知処理等を、汎用アプリケーションで作成した文書に対して行うことが可能となる。

【0062】図6は、ジョブ記述ファイル18の内容例を示す図である。図6において、ジョブ記述ファイル18は、ジョブ記述ファイルヘッダ61、DEVMODE情報62及びジョブ記述ファイルフッタ63を備えている。なお、これらの情報は、Windowsシステム概値である。DEVMODE情報62は、仮想ドライバ14のDEVMODE構造体情報によって作成される。また、DEVMODE情報62には、プリンタドライバプライベート情報64が仮想ドライバ14により付加される。DEVMODE情報62として、用紙サイズ、用紙種、用紙方向などがあり、プリンタドライバプライベート情報64として、出力制御ファイル名などがある。

【0063】図7は、出力制御ファイル17の内容例を示す図である。図7において、出力制御ファイル17は、CAPCAL出力制御ファイルヘッダ71及び出力制御情報75を備え、出力制御情報75は、動作モード情報72、文書付加情報73及び文書属性情報74を備えている。CAPCAL出力制御ファイルヘッダ71は、CAPCAL出力制御ファイル17の情報を含み、バージョン情報などが格納されている。動作モード情報72は、動作モード(通常モード/配布モード/分散モード)が格納されている。また、仮想ドライバ(CAPCALドライバ)14により指定される配布先情報を含み、配布モード(自動配布/自動配布)、配布先の名称、検索ページ、検索モード(区切りモード/修飾モード/属性モード/サイズモード/色モード/矩形モード)、修飾モードの修飾の種類(下線/網掛け)、属性モードの属性種類(標準/斜体/太字/太字斜体)、サイズモードの文字サイズ、色モードの色コード、及び矩形モードの領域などが格納されている。

【0064】文書付加情報73は、仮想ドライバ14により指定される文書の付加情報を含み、フォーム付加フラグ、バナー付加フラグ、フォームファイルの場所、フォームファイル名、及びバナー挨拶文などが格納されている。

【0065】文書属性情報74は、仮想ドライバ14により指定される文書の属性情報を含み、パスワードフラグ、出力モード(即印刷/日時指定印刷/時間指定印刷/遅延印刷/印刷しない)、日時指定印刷の指定日時、時間指定印刷の指定時間、保存モード(日時指定保存/時間指定保存/無条件保存/保存しない)、及び日時指定保存の指定時間などが格納されている。

【0066】図8は、仮想ドライバ14の動作を示すフローチャートであり、仮想ドライバ14によりジョブ記述ファイル18と出力制御ファイル17とを作成する処理を示すものである。この処理では、ユーザがCAPCALプリンタドライバプロパティを操作するフェーズ(ステップS1～S3)と、仮想ドライバ14の印刷フェーズ(ステップS4～S6)とが設けられている。

【0067】図8において、まず、一般のWindowsプリンタドライバと同様に、CAPCALプリンタドライバプロパティを開く(ステップS1)。なお、CAPCALプリンタドライバプロパティは、プリントフォルダやアプリケーションから開くことが可能である。

【0068】次に、CAPCALプリンタドライバプロパティが開かれると、ドライバ出力制御設定画面の表示に従って出力制御情報75を設定し、DEVMODE情報62及び出力制御情報75をレジストリに保存する(ステップS2)。そして、アプリケーション1より印刷ダイアログを開き、印刷を実行する(ステップS3)。

【0069】次に、仮想ドライバ14は、CAPCAL

プリンタドライバプロパティからDEV MODE情報62及び出力制御情報75を取得する(ステップS4)。なお、CAPCALプリンタドライバプロパティにDEV MODE情報62及び出力制御情報75が保存されていなければ、仮想ドライバ14に設定されている値を使う。

【0070】次に、仮想ドライバ14は、CAPCALプリンタドライバプロパティから取得したDEV MODE情報(出力制御ファイル名を含む)62をGDI2に渡す(ステップS5)。

【0071】次に、仮想ドライバ14は、CAPCALプリンタドライバプロパティから取得した出力制御情報75を、DEV MODE情報62に付加されている出力制御ファイル名で出力制御ファイル17に書き込む(ステップS6)。

【0072】このように、仮想ドライバ14が、プリンタ機種やメーカーに依存しない共通な情報(DEV MODE情報62)をWindowsシステム(GDI2)に返し、ジョブ記述ファイル18をGDI2に作成させることにより、印刷に必要な情報(用紙種、給紙方式、用紙方向等)を与えるようにしている。また、仮想ドライバ14が、出力制御ファイル17を独自に作成し、配布分散処理に必要な出力制御情報75をCAPCALプリンタプロセッサ21に与えている。このため、汎用のアプリケーション1で作成した文書について、メーカーやプリンタ機種に依存しない配布印刷処理を実現することが可能となる。すなわち、ユーザは、ネットワーク上のどのプリンタに印刷させる場合も、仮想ドライバ14で印刷を実行すれば良く、仮想ドライバ14が与える情報によって、プリンタ機種やメーカーで特定した機能を意識することなく、ネットワーク上の多種多様なプリンタ9を利用できる。

【0073】さらに、アプリケーション1で作成したドキュメントは、出力装置(プリンタ9)の物理仕様に依存せず、仕様が異なるプリンタに出力しても期待する結果を得ることが可能となることから、可搬性を向上させることができる。この結果、システム上の全てのアプリケーション1の出力ドキュメントを処理対象として、物理的な出力装置の個々の仕様に依存しない論理デバイスドライバ方式を提供することが可能となる。また、出力装置の機能が不足する場合も、適切なエミュレートで期待する出力を得ることができる。

【0074】図9は、DEV MODE情報の設定画面を示す図である。CAPCALプリンタドライバプロパティを開いた際に、このDEV MODE情報の設定画面を表示させることにより、給紙位置、用紙サイズ、印刷用紙、拡大/縮小、用紙方向、紙種、コピー枚数などを入力することができ、図6のDEV MODE情報62を設定することができる。

【0075】図10は、自動配布情報の設定画面を示す

図である。CAPCALプリンタドライバプロパティを開いた際に、この自動配布情報の設定画面を表示させることにより、検索ページ、印刷時指定、検索モード(区切りモード/修飾モード/サイズモード/色モード/属性モード)、検索範囲などを入力することができ、図7の出力制御情報75を設定することができる。自動配布とは、ユーザが配布先を指定しなくても、配布される文書の中から配布先を自動的に抽出して配布を行うものである。ここで、検索ページは、配布先を抽出する文書のページを示す。区切りモードでは、指定された文字列で挟まれた領域が検索範囲とみなされ、例えば、括弧を指定すると、括弧で挟まれた文字列が配布先となる。修飾モードでは、修飾された文字列が配布先として検出され、サイズモードでは、指定された大きさを有する文字列が配布先とみなされ、色モードでは、指定された色を有する文字列が配布先とみなされ、属性モードでは、指定された属性の文字列が配布先とみなされる。

【0076】また、検索範囲の入力は、この検索範囲を1枚の仮想的な用紙とみなし、用紙を分割して得られた領域を指定することにより、検索範囲を特定するものである。例えば、縦横の中心線で仮想用紙を4分割した場合の右上の領域と指定する。そして、この仮想用紙での領域の指定を実際用紙に変換する場合、縦横の中心線で実際の用紙を4分割し、4分割して得られた右上の領域を検索範囲とする。なお、用紙の分割の方法は、4分割の他に、8分割や16分割などであってもよい。この検索範囲の入力により、システム上の全てのアプリケーションの出力ドキュメントを処理対象として、アプリケーションから独立した処理でドキュメントの領域を指定可能となるとともに、ドキュメントの形状(用紙サイズ、用紙の置き方)に依存することなくドキュメントの領域を指定できる。

【0077】図11(a)は、本発明の一実施例に係わる領域指定制御部を説明するブロック図である。図11(a)において、仮想ドライバ14のユーザインタフェース81には、領域指定制御部82が設けられている。領域指定制御部82は、ユーザが指定した領域情報を仮想ドライバ14によって取り込み、文書の印刷情報として登録する。

【0078】図11(b)は、実際の用紙を表す長方形を表示することにより、ユーザに領域指定させる方法を説明する図である。図11(b)において、用紙の置き方に依存して、ポートレートの用紙83を表す長方形と、ランドスケープの用紙85を表す長方形とが表示される。そして、ユーザは、ポートレートの用紙83及びランドスケープの用紙85のそれぞれに対し、検索範囲を示す領域84、86を指定する。

【0079】図11(c)は、仮想用紙87を表す長方形を表示することにより、ユーザに領域指定させる方法を説明する図である。図11(c)において、用紙を表

す長方形を、正方形の仮想用紙87としてMMI（マンマシンインターフェース）としている。ユーザは、この仮想用紙87上で矩形領域88を指定する。

【0080】そして、印刷を実行するときに、仮想用紙87上で指定された情報を取り込み、ポートレート用の用紙89、及びランドスケープ用の用紙91のそれぞれの場合について、仮想用紙87上の矩形領域88を実際用の用紙に合わせて変換する。仮想用紙87の大きさを、実際の用紙の大きさに対応させて変換する場合、例えば、仮想用紙87の縦横の変倍率に対応させて、仮想用紙87に指定された矩形領域88を変換する。このことにより、仮想用紙87上での矩形領域88の相対的な位置を保存したまま、実際の用紙89、91上での検索範囲を示す領域90、92に変換することができる。

【0081】例えば、ポートレート用の用紙89及びランドスケープ用の用紙91のいずれの場合においても、配布先は、用紙89、91の上方の欄に記入されることが多い。このため、仮想用紙87上の上方の領域を矩形領域88として指定することにより、仮想用紙87がポートレート用の用紙89及びランドスケープ用の用紙91のいずれに変換された場合においても、検索範囲を示す領域90、92が用紙89、91の上方の欄に位置するように変換され、検索範囲の指定を効率的に行うことが可能となる。

【0082】このように、領域指定機能を、アプリケーション及びドキュメントから独立した階層で提供し、領域指定のMMI（マンマシンインターフェース）を、ドキュメントの形状から独立した正方形の仮想用紙87としたことにより、配布先の検索などのため印刷ドキュメントの領域を指定することが必要な場合、アプリケーションに領域指定機能を組み込むことなく、全ての出力ドキュメントに対して領域設定ができる。この領域設定は、全てのアプリケーションから出力される全てのドキュメントに適用できる。また、領域設定は、ドキュメントから独立しており、1つの領域設定で全ての出力ドキュメントを対象とすることができる。さらに、各ドキュメントの形状（用紙サイズ、置き方）に依存せずに設定することができるので、用紙サイズ、置き方の違いによる領域指定作業の軽減が図れるとともに、一度設定すれば、ドキュメントの形状を変更しても再設定を不要とすることができる。

【0083】図12は、指定配布情報の設定画面を示す図である。CAPCALプリンタドライバプロパティを開いた際に、この指定配布情報の設定画面を表示させることにより、配布先モード（配布先名指定／グループ名指定／プリンタ名指定）、配布先名、グループ名、プリンタ名などを入力することができ、図7の出力制御情報75を設定することができる。例えば、配布先名指定を選択した場合、「東京総務課」などの論理名で印刷物の配布先を指定することが可能となり、プリンタ名を意識

することなく配布先を指定することが可能となる。また、グループ名指定を選択した場合、1回の指定で複数の配布先を選択することが可能となり、配布先の選択の手間を軽減することが可能となる。さらに、選択された配布先を、画面上で自由に追加したり、削除したりすることが可能となり、配布先の変更も容易に行うことが可能となる。

【0084】図13は、本発明の一実施例に係わる分散印刷情報の設定画面を示す図である。CAPCALプリンタドライバプロパティを開いた際に、この分散印刷情報の設定画面を表示させることにより、分散方法（台数均等／部数別／種別／マニュアル指定）、カラーとモノクロの区別、キューイングの数、プリンタのエラーなどを入力することができ、図7出力制御情報75を設定することができる。分散方法を指定することにより、印刷部数を複数のプリンタに均等に割り振ったり、プリンタの印刷能力や稼働状況に応じて割り振ったりすることが可能となる。また、カラーとモノクロの区別を指定することにより、カラープリンタには、カラーのページだけを出力させたり、モノクロプリンタには、モノクロのページだけを出力させたりすることが可能となる。キューイングの数を指定することにより、出力待ちが貯まっているプリンタを避けて、空いているプリンタを優先させて出力させることができる。プリンタのエラーを指定することにより、配布分散スレッド37がプリンタを決定する時に、エラーが発生しているプリンタを分散対象のプリンタとして扱わないようにすることができる。

【0085】図14は、CAPCALプリントプロセス21の構成を示すブロック図である。図14において、印刷データ読み込み制御部101を介して読み込まれた印刷データは、印刷データ格納部102に格納される。印刷データ格納部102に格納された印刷データは、印刷データ構成部103で各ページごとにデータ区分され、CAPCALスプールファイル生成部104に送られる。CAPCALスプールファイル生成部104は、印刷データの各頁ごとの出力による頁管理情報を頁管理情報格納部105に格納し、CAPCALスプールファイル57のEMFファイル24に各頁の印刷データを出力する。全ての印刷データがCAPCALスプールファイル57のEMFファイル24に出力されると、CAPCALジョブ記述ファイル23が作成される。

【0086】CAPCALジョブ記述ファイル23を作成する場合、DEV MODE情報とWindowsジョブ情報とが、印刷ジョブ情報読み込み制御部106から読み込まれ、CAPCAL出力制御情報が、CAPCAL出力制御ファイル読み込み制御部107から読み込まれ、CAPCALユーザ情報が、CAPCALユーザ情報読み込み制御部108から読み込まれる。そして、DEV MODE情報、Windowsジョブ情報、CAPCAL出力制御情報及びCAPCALユーザ情報が、C

APCALジョブ情報格納部109に格納される。印刷ジョブ情報構成部110は、CAPCALジョブ情報格納部109に格納された情報に基づいて、印刷ジョブ情報を構成し、CAPCALスプールファイル生成部105を介してCAPCALスプールファイル57のCAPCALジョブ記述ファイル23に出力する。この際、CAPCALスプールファイル生成部105は、頁管理情報格納部105に格納されている頁管理情報を、印刷ジョブ情報構成部110で構成された印刷ジョブ情報に付加し、CAPCALジョブ記述ファイル23に出力する。

【0087】図15は、CAPCALジョブ記述ファイル23の内容例を示す図である。図15において、CAPCALジョブ記述ファイル23は、CAPCALジョブ記述ファイルヘッダ111、Windowsジョブ情報112、DEVMODE情報113、CAPCAL出力制御情報114、CAPCALユーザ情報115及び印刷データ頁管理情報116を備えている。

【0088】CAPCALジョブ記述ファイルヘッダ111は、CAPCALジョブ記述ファイル23のヘッダ情報として、バージョン番号などの情報を含んでいる。Windowsジョブ情報112は、Windowsのジョブ情報として得られる情報であり、印刷を依頼したユーザ名、印刷ジョブの総頁数、印刷ジョブが発生したホスト名、印刷ジョブタイトル名、印刷ジョブ発生時刻などの情報を含んでいる。

【0089】DEVMODE情報113は、アプリケーション1及び仮想ドライバ14により指定される印刷ジョブのDEVMODE情報113である。CAPCAL出力制御情報114は、図7の出力制御情報75と同一の内容を有している。

【0090】CAPCALユーザ情報115は、CAPCAL用に定義されたユーザの情報であり、氏名、会社名、部署名、電話番号、FAX番号、パスワードなどの情報を含んでいる。

【0091】印刷データ頁管理情報116は、印刷データのファイル上の各頁の先頭位置を示すインデックス情報である。この印刷データ頁管理情報116を参照することにより、配布印刷や分散印刷を容易に行うことができる。

【0092】図16は、CAPCALプリントプロセッサ21の動作を示すフローチャートである。図16において、まず、CAPCALプリントプロセッサ21は、EMFファイル24の作成を行う（ステップS11～S14）。具体的には、印刷データ読み込み制御部101を介して読み込まれた印刷データを、印刷データ格納部102に格納する（ステップS11）。次に、印刷データ構成部103は、印刷データ格納部102から、1頁分の印刷データを描画単位ごとに取得し、CAPCALスプールファイル生成部104を介して、CAPCAL

スプールファイル57に各頁の印刷データを出力する（ステップS12）。この際、CAPCALスプールファイル生成部104の頁管理情報格納部105には、印刷データ頁管理情報116が格納される。次に、未処理の印刷データがあるかどうかを判断し（ステップS13）、未処理の印刷データがある場合、ステップS12に戻って処理を繰り返す。次に、未読み込みのスプールデータがあるかどうかを判断し（ステップS14）、未読み込みのスプールデータがある場合、ステップS11に戻って処理を繰り返す。

【0093】CAPCALプリントプロセッサ21が、CAPCALスプールファイル57に印刷データを出力した後、CAPCALジョブ記述ファイル23の作成を行う（ステップS15～S17）。具体的には、印刷ジョブ情報読み込み制御部106によりWindowsジョブ情報112とDEVMODE情報113とが読み込まれ、CAPCAL出力制御ファイル読み込み制御部107によりCAPCAL出力制御情報114が読み込まれ、CAPCALユーザ情報読み込み制御部108によりCAPCALユーザ情報115が読み込まれる。そして、Windowsジョブ情報112、DEVMODE情報113、CAPCAL出力制御情報114及びCAPCALユーザ情報115が、CAPCALジョブ記述ファイル23生成用の情報として、CAPCALジョブ情報格納部109に格納される（ステップS15）。

【0094】次に、印刷データの各頁ごとの書き込みにより頁管理情報格納部105に格納された印刷データ頁管理情報116を、CAPCALスプールファイル57に書き込む（ステップS16）。

【0095】次に、CAPCALジョブ情報構成部110は、CAPCALジョブ情報格納部109に格納されているWindowsジョブ情報112、DEVMODE情報113、CAPCAL出力制御情報114及びCAPCALユーザ情報115を、CAPCALスプールファイル生成部104を介してCAPCALスプールファイル57に書き込む。

【0096】このように、CAPCALプリントプロセッサ21は、Windowsシステムの印刷制御に要求された印刷データ（スプールファイル）をプリンタ9に印刷出力させずに、CAPCAL制御の印刷データとして独自にスプールすることにより、Windowsシステムの印刷制御とは全く異なるサービスの提供が可能となる。この結果、ドキュメントの頁単位、あるいはドキュメント単位に、様々なサービス（出力先の操作／出力順の操作など）をユーザに提供することが可能となる。

【0097】ここで、印刷データの作成については、Windowsシステムの印刷制御を通して行っているため、汎用アプリケーションからの印刷についても、独自の印刷制御のサービスを提供することが可能となることから、システム上の全てのアプリケーションの出力ドク

ュメントを処理対象として、例えば、出力先の追加・変更、頁単位での操作、出力順の操作など様々なサービスを提供することが可能となる。

【0098】次に、本発明の実施例に係わる配布印刷処理について説明する。配布印刷処理は、図3の配布分散処理スレッド37が行う。配布分散処理スレッド37では、CAPCALジョブ記述ファイル23中の出力制御情報75で指示された内容に従って、配布印刷、分散印刷、通常印刷のそれぞれの動作が行われる。配布印刷には、自動配布及び指定配布の2種類がある。自動配布は、配布先を印刷ジョブの内容により自動的に決定するモードである。指定配布は、印刷ジョブの配布先をユーザが予め指定するモードである。これら2つのモードのうち、どちらの動作を行うかは、CAPCALジョブ記述ファイル23中で定義することができる。

【0099】図17は、本発明の一実施例に係わる自動配布処理を説明するブロック図であり、自動配布印刷における配布先一覧を得るまでの概要を示す。図17において、サーバ301が、クライアント300から送られてきたCAPCALジョブ記述ファイル23及びEMFファイル24の内容を受信すると、これらの内容をCAPCALスプールドキュメントキュー33のCAPCALジョブ記述ファイル34及びEMFファイル35に格納する。配布分散処理スレッド37は、CAPCALジョブ記述ファイル34の内容を読み出し、配布先候補としての文字列122を印刷ジョブ121から検索するための検索条件124を、CAPCALジョブ記述ファイル34から取得する。この検索条件124には、検索ページ及び検索モード（区切り文字／修飾／属性／サイズ／色／矩形領域内）などがあり、この検索条件124を指定することにより、印刷ジョブ121から配布先候補を抽出する方法を指定することができる。なお、この検索条件124は、クライアント300側において、CAPCALプリンタドライバプロパティを開き、図10の自動配布情報の設定画面を表示させることにより、設定することができる。

【0100】配布分散処理スレッド37は、この検索条件124を元に印刷ジョブ121の検索を行い、検索条件124に適合した文字列122を配布先名として取得するとともに、レジストリ123から配布先定義を取得する。レジストリ123には、配布先定義として、N台分配布先名と出力装置名との対応関係が格納されている。このため、比較部125において、印刷ジョブ121から取得した配布先名と、レジストリ123に定義された配布先名とを比較し、レジストリ123に定義された配布先名と一致する文字列122だけを選択することにより、配布すべき全ての出力装置を自動的に決定することができる。

【0101】図18は、自動配布時における検索条件例を示す図である。図18において、検索モードには、

「区切り文字」、「修飾」、「属性」、「サイズ」、「色」、「矩形領域内」などがある。「区切り文字」は、指定された始端文字列と終端文字列との間の文字列122を配布先候補として抽出するものである。なお、印刷時指定により、区切り文字をそのまま出力させたり、スペースで置き換えて出力させたりすることができる。「修飾」は、下線や網掛けなどにより修飾された文字列122を配布先候補として抽出するものである。なお、印刷時指定により、修飾文字をそのまま出力させたり、修飾を解除して出力させたりすることができる。「属性」は、標準、斜体、太字、太字斜体などの属性を有する文字列122を配布先候補として抽出するものである。なお、印刷時指定により、属性が与えられた文字をそのまま出力させたり、属性を解除して出力させたりすることができる。「サイズ」は、指定された文字サイズ（ポイント数）を有する文字列122を配布先候補として抽出するものである。「色」は、指定された色コードを有する文字列122を配布先候補として抽出するものである。なお、印刷時指定により、指定された色コードを有する文字列をそのままの色で出力させたり、黒に変換して出力させたりすることができる。「矩形領域内」は、検索領域として指定された矩形内の文字列122を配布先候補として抽出するものである。

【0102】図19は、本発明の一実施例に係わる自動配布処理の動作を示すフローチャートであり、配布分散スレッド37が出力装置名称を決定する概要を示す。図19において、まず、配布分散スレッド37は、CAPCALジョブ記述ファイル34の出力制御情報114から検索条件124を取得する。検索条件124として取得するのは、検索ページ、検索モード、条件詳細である（ステップS21）。検索ページは、検索を行う印刷ジョブ121のページである。検索モードは、検索ページ中のどのデータを検索対象とするかを指定するものである。条件詳細は、図18に示すように、指定された検索条件124により異なる情報が格納されている。

【0103】次に、取得した検索条件124に基づき、印刷ジョブ121の検索を行い、検索条件124に該当する全文字列122を印刷ジョブ121から取得し、配布先の候補とする（ステップS22）。次に、レジストリ123から配布先定義（全配布先）を取得し、配布先定義情報一覧を作成する（ステップS23）。配布先定義には、論理的な配布先名称とそれに対応する物理的な出力装置名称とが対となった情報が、配布対象となる出力先の分だけ登録されている。次に、作成された配布先定義情報一覧の中から、配布先定義を1件だけ取り出す（ステップS24）。

【0104】次に、印刷ジョブ121から取得した配布先候補と、配布先定義から取得した配布先名称とを比較し（ステップS25）、印刷ジョブ121から取得した配布先候補と、配布先定義から取得した配布先名称とが

一致する場合、対応する出力装置名称を出力先一覧テーブルに追加する(ステップS26)。

【0105】次に、配布先定義情報一覧に比較していない配布先定義があれば(ステップS27)、ステップS24に戻り、配布先定義情報一覧から新たな配布先定義を取り出し、比較処理を行う。そして、配布先定義情報一覧の中の配布先定義を全て比較した場合、配布先決定処理を終了する。配布先決定処理が終了すると、作成した出力先一覧を元に、それぞれの出力先に印刷ジョブ121の投入を行い、配布印刷を行う。

【0106】このように、クライアント300側で指定された配布先の検索条件124を、サーバ301側のCAPCALジョブ記述ファイル34の内容として送信し、クライアント300から送られた検索条件124に基づいて、ドキュメントに含まれる予め指定された文字列を自動的に判別することにより、システム上の全てのアプリケーション1の出力ドキュメントを対象として、印刷ジョブ121から配布先を自動的に検出することが可能となる。この結果、印刷ドキュメントごとに配布先を指定する手間を省くことが可能となり、システム環境を(配布先に対応するプリンタ名称)を意識することなく、1回の印刷指示により検索条件124に該当する全ての配布先へ自動的に印刷を行うことが可能となる。さらに、ユーザの配布方法の選択に幅を持たせることが可能となり、自由度の高い配布処理を実現することが可能となる。

【0107】図20は、本発明の一実施例に係わる指定配布処理を説明するブロック図であり、指定配布印刷における配布先を得るまでの概要を示す。図20において、サーバ301が、クライアント300から送られてきたCAPCALジョブ記述ファイル23及びEMFファイル24の内容を受信すると、これらの内容をCAPCALスプールドキュメントキュー33のCAPCALジョブ記述ファイル34及びEMFファイル35に格納する。配布分散処理スレッド37は、CAPCALジョブ記述ファイル34の内容を読み出し、ここから配布先情報131を取得する。配布先情報131は、指定配布モードとN個の配布先名称とを備え、指定配布モードには、グループ指定、論理名指定(配布先名指定)及び物理名指定(プリンタ名指定)があり、配布先名称には、グループ名、論理名(プリンタ名)及び物理名(配布先名)がある。なお、この配布先情報131は、クライアント300側において、CAPCALプリンタドライバプロパティを開き、図11の指定配布情報の設定画面を表示させることにより、設定することができる。

【0108】ここで、配布分散処理スレッド37は、配布先情報131の指定配布モードが物理名指定の場合、配布先情報131の配布先名称の物理名から、配布すべき出力装置を決定する。

【0109】一方、配布先情報131の指定配布モード

がグループ名若しくは論理名である場合、配布分散処理スレッド37は、レジストリ132から、配布先定義情報及びグループ定義情報を取得する。レジストリ132には、配布先定義として、N台分の配布先名と出力装置名との対応関係が格納されているとともに、グループ定義として、グループ名及び配布先へのリンク情報が格納されている。このため、配布分散処理スレッド37は、レジストリ132に定義されている情報を基に、配布すべき出力装置を決定することができる。

【0110】図21は、本発明の一実施例に係わる指定配布処理の動作を示すフローチャートであり、指定配布において配布分散スレッド37が出力装置名称を決定するまでの概要を示す。

【0111】図21において、配布分散処理スレッド37は、CAPCALジョブ記述ファイル34の出力制御情報114から、配布先情報131として、指定配布モードと配布先名称とを取得する(ステップS31)。

【0112】次に、配布分散処理スレッド37は、レジストリ132から、配布先定義情報とグループ定義情報を取得する(ステップS32)。配布先定義には、N台分の配布先名と出力装置名との対応関係が登録されている。グループ定義には、論理的なグループ名称とそれに対応する配布先定義へのリンクが登録されている。

【0113】次に、配布先情報131から取得した指定配布モードが「物理名指定」である場合(ステップS33)、配布先情報131で示される配布先名称はそのまま出力先名称となる。このため、配布先情報131の配布先名称をそのまま出力先として、出力先一覧を作成する(ステップS34)。一方、配布先情報131から取得した指定配布モードが「論理名指定」である場合(ステップS35)、配布先情報131から取得した配布先名称は論理名称である。このため、レジストリ132から取得した配布先定義情報を参照することにより出力装置を決定し、出力先一覧を作成する(ステップS36)。

【0114】また、配布先情報131から取得した指定配布モードが「グループ指定」である場合、配布先情報131から取得した配布先名称はグループ名称である。このため、レジストリ132から取得したグループ定義情報と配布先定義情報を参照することにより出力装置を決定し、出力先一覧を作成する(ステップS37)。出力先一覧の作成が終了すると、作成した出力先一覧を元にそれぞれの出力先に印刷ジョブの投入を行い、配布印刷を行う。

【0115】このように、クライアント300側で指定された配布先情報131を、サーバ301側のCAPCALジョブ記述ファイル34の内容として送信し、クライアント300から送られた配布先情報131に基づいて、配布先を決定することにより、システム上の全てのアプリケーション1の出力ドキュメントを対象として、

配布出力先をドキュメント単位に任意に指定することが可能となる。

【0116】また、レジストリ123、132に、N台分の配布先名と出力装置との対応関係を登録しておくことにより、1つの論理的な配布先に複数の出力装置を定義することが可能となり、配布先を1つ決定するだけで、複数の出力装置にドキュメントを配布することが可能となるとともに、配布先を物理的な出力装置ではなく、論理的な名称で指定可能となる。このため、希望する配布先を印刷実行時に複数選択したり、印刷ドキュメント毎に配布先を指定したり、配布対象であるプリンタの名称を全て指定したりする手間を省くことが可能となり、配布先に対応するプリンタ名称を意識することなく、1回の印刷指示で条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0117】また、利用者が自由に配布先決定方式を選択することが可能となり、ドキュメントの内容を解析することにより、配布先を印刷ドキュメントデータから自動的に決定したり、配布先対象とする文字列を直接指定することにより、配布先をドキュメントに対して指定したりすることが可能となる。

【0118】さらに、ドキュメント内容による確実な配布先指定が可能となり、配布先の一時的な変更（配布停止など）を容易、かつ安全に行うことが可能となる。この結果、ユーザによる配布方法選択に幅がある自由度の高い配布処理が可能となり、自由度が高く運用状況に柔軟に対応することができるシステム構築及び運用が可能となる。

【0119】次に、本発明の実施例に係わるドキュメント出力監視処理方式について説明する。図22は、本発明の一実施例に係わるドキュメント出力監視処理部の構成を示すブロック図である。図22において、印刷を実行したことにより、スプールファイル142がクライアント300に生成される。そして、クライアント300に生成されたスプールファイル142の内容が、転送モジュール143を介してサーバ301へ転送される。この際、転送モジュール143は、スプールファイル142の内容をサーバ301へ転送したことを、クライアント送信ログ144に出力する。

【0120】サーバ301は、転送モジュール143から送信されたスプールファイル142の内容を受信モジュール148を介して受け取ると、サーバ301上にスプールファイル147を生成する。この際、受信モジュール148は、サーバ301上にスプールファイル147を生成したことを、サーバ受信ログ149に出力する。

【0121】出力制御モジュール150は、スプールファイル147がサーバ301上に生成されると、出力を開始することを出力開始終了ログ151へ書き出し、印刷制御モジュール152にプリンタでの印刷を開始させ

る。プリンタでの印刷が完了すると、印刷制御モジュール152は、プリンタでの印刷が完了したことを印刷ログ153に出力し、出力制御モジュール150へ印刷完了を通知する。その通知を受けた出力制御モジュール150は、出力が完了したことを出力開始終了ログ151へ書き出す。

【0122】通常の印刷では、上述した処理により、クライアント300からサーバ301を経由したプリンタへの印刷は完了する。一方、クライアント300から指定されたプリンタが他サーバ304に対するものであった場合には、サーバ301から他サーバ304に対し、スプールファイル147の内容の転送が行われる。

【0123】印刷するプリンタが他サーバ304の管轄の場合、サーバ301の出力制御モジュール150は、スプールファイル147の内容を転送モジュール154を介して他サーバ304へ転送する。この際、転送モジュール154は、スプールファイル147の内容を他サーバ304へ転送したことを、サーバ送信ログ155に出力する。

【0124】他サーバ304の受信モジュール158は、サーバ301の転送モジュール154から送信されたスプールファイル147の内容を受け取ると、他サーバ304上にスプールファイル157を生成する。この際、受信モジュール158は、サーバ301からスプールファイル147の内容を受け取ったことを、サーバ受信ログ159に出力する。

【0125】他サーバ304の出力制御モジュール160は、スプールファイル157が生成されると、出力を開始することを出力開始終了ログ161へ書き出し、印刷制御モジュール162にプリンタでの印刷を開始させる。プリンタでの印刷が完了すると、印刷制御モジュール162は、プリンタでの印刷が完了したことを印刷ログ163に出力し、出力制御モジュール160へ印刷完了を通知する。その通知を受けた出力制御モジュール160は、出力が完了したことを出力開始終了ログへ161書き出す。

【0126】クライアント300からサーバ301への転送及び印刷までの過程は、サーバ301から他サーバ304への転送及び印刷までの過程と同じ制御となる。また、例えば、WANに接続されているサーバ間の転送等、支店から本店のプリンタへ向けて印刷を行う場合において、送受信が可能な限られているマシン間でスプールファイルを転送し、そこから指定のプリンタに印刷可能なサーバへの転送が行われることがある。このように、他サーバ304から別のサーバへの転送がさらに発生する場合は、上述した転送処理が繰り返し行われる。

【0127】図22の処理では、クライアント300→サーバ301→他サーバ304におけるスプールファイル142、147、157の処理過程（結果）を示すログが、クライアント送信ログ144、サーバ受信ログ1

49、出力開始終了ログ151、印刷ログ153、サーバ送信ログ155、サーバ受信ログ159、出力開始終了ログ161、印刷ログ163に出力されたことになる。ログ追跡モジュール145が、これらの個々のログから、これらの順番に情報を取得し、1回の印刷実行に対する情報に関連付けることにより、異なるマシン間での印刷の開始から終了までの過程を全て追尾することが可能となる。

【0128】このように、クライアント300での印刷開始、サーバ301、304での印刷実行までの経過、及び印刷実行結果をそれぞれログに格納し、ユーザが認識可能であるドキュメント出力処理をキーとして、それらの情報が格納された複数のログの情報表示や操作などの処理を可能としている。このことにより、出力装置を制御するサーバがネットワーク上に複数存在するシステムにおいて、利用者が投入した複数の出力装置に対する複数のドキュメント出力処理を、出力装置ではなくドキュメント出力処理を主体に監視(状態監視、結果監視)することが可能となる。この結果、ユーザは、自分が投入した印刷処理を、リアルタイムで監視することが可能となるとともに、エラー発生時に、プリンタを変更したり、頁を指定しての再印刷を行わせたりする指示を、簡単、安全、かつ的確に行うことが可能となる。

【0129】図23は、ログ情報の追跡の流れを示すブロック図である。図23において、クライアント300によりクライアント送信ログ144が生成され、サーバ301によりサーバ受信ログ149、出力開始終了ログ151、印刷ログ153及びサーバ送信ログ155が生成され、他サーバ304によりサーバ受信ログ159、出力開始終了ログ161及び印刷ログ163が生成されている。そして、ログ追跡モジュール145により、クライアント送信ログ144、サーバ受信ログ149、出力開始終了ログ151、印刷ログ153、サーバ送信ログ155、サーバ受信ログ159、出力開始終了ログ161及び印刷ログ163が順次に追跡される。

【0130】図24は、ログファイルに収めるログ情報の内容例を示す図である。図22及び図23には、クライアント送信ログ144、サーバ受信ログ149、出力開始終了ログ151、印刷ログ153、サーバ送信ログ155、サーバ受信ログ159、出力開始終了ログ161及び印刷ログ163の8種類のログが存在するが、これらのログ情報の構造は、以下に示すように、全て共通となっている。

【0131】図24において、「ログ情報の種類」170には、ログ情報の種類を示すコードを格納する。例えば、クライアント送信ログ144には、「ログ情報の種類」170として送信ログ(クライアント)を示すコードを格納し、サーバ受信ログ149、159には、「ログ情報の種類」170として受信ログ(サーバ)を示すコードを格納し、出力開始終了ログ151、161に

は、「ログ情報の種類」170として出力開始終了ログ(サーバ)を示すコードを格納し、印刷ログ153、163には、「ログ情報の種類」170として印刷ログ(サーバ)を示すコードを格納し、サーバ送信ログ155には、「ログ情報の種類」170として送信ログ(サーバ)を示すコードを格納する。

【0132】「サーバ上の識別子」171には、サーバ上でユニークに割り当てられる通し番号を格納する。なお、サーバ上の識別子171は、サーバが受信したときに割り当てられる番号である。

【0133】「クライアント上の識別子」172には、印刷を実行したクライアント上でユニークに割り当てられる通し番号を格納する。「クライアントのホスト名」173には、印刷を実行したクライアントのホスト名称を格納する。

【0134】「出力先名(プリンタ等)」174には、印刷の出力先の名称を格納する。例えば、「出力先名(プリンタ等)」174にプリンタ名が格納されると、サーバ経由で印刷が行われる。また、「出力先名(プリンタ等)」174に他サーバ管轄のプリンタが格納されると、サーバ間を経由した印刷が行われる。

【0135】「出力情報」175には、印刷実行時に指定した印刷物固有のジョブタイトル名(文書や表のドキュメント名)177、ユーザ名178、用紙サイズ179、出力部数180、出力頁順181、出力開始頁番号182、出力終了頁番号183、出力終了日時184、出力結果コード185などが格納される。この「出力情報」175は、印刷物の内容を確認する際の表示用の情報として使用される。

【0136】「制御情報」176には、通常/配布/分散等の印刷の種類を示す動作モード186や、通常/日時/遅延等の印刷出力のタイミングを示す出力モード187や保存モード188などが格納される。「制御情報」176は、印刷の実行を制御する情報として使用され、どのような制御下で印刷が行われたかを確認する際に有効となる。

【0137】なお、異なるマシン間でのログ情報は、「クライアント上の識別子」172と「クライアントのホスト名」173とのセットに関連付けることができる。このように、異なる処理でも同一構造のログ情報に格納することにより、処理結果の記憶装置への入出力を共通に扱うことが可能となる。

【0138】図25は、本発明の一実施例に係わるドキュメント出力監視方法を示すフローチャートである。図25において、ログ追跡モジュール145は、ログ追跡を開始するクライアント300上のクライアント送信ログ144からログ情報を獲得する(ステップS41)。

【0139】次に、印刷の実行によりスプールファイル142を転送する送信先のサーバ301のサーバ名称を、クライアント送信ログ144のログ情報から獲得す

る(ステップS42)。

【0140】次に、同じクライアント300のホスト名173と識別子172を持つログ情報を、送信先サーバ301上のサーバ受信ログ149から獲得する(ステップS43)。

【0141】次に、サーバ301の出力開始終了ログ151から、出力開始情報を獲得する(ステップS44)。次に、同じクライアント300のホスト名173と識別子172を持つログ情報を以降のログの中から検索する際の検索開始日時として、出力開始終了ログ151から獲得した出力開始情報の出力終了日時184をセットする(ステップS45)。

【0142】次に、出力開始終了ログ151から、出力終了情報を獲得する(ステップS46)。この出力終了情報は、出力が完了している場合にのみ取得することが可能である。

【0143】次に、出力開始終了ログ151のログ情報の中に出力終了情報が有るならば(ステップS47)、検索終了日時として、出力開始終了ログ151から獲得した出力終了情報の出力終了日時をセットする(ステップS48)。

【0144】次に、出力開始終了ログ151から獲得した出力開始情報の出力先名174がプリンタの場合(ステップS49)、同じプリンタ(出力先)で、かつ同じクライアント300のホスト名173と識別子172を持ち、検索開始日時と検索終了日時との間の出力終了日時184を格納しているログ情報を、印刷ログ153から検索する(ステップS50)。

【0145】サーバ301からプリンタへの印刷が完了していれば(ステップS51)、以上の処理によりログの追尾は終了する。出力開始終了ログ151から獲得した出力開始情報の出力先名174がプリンタでなく、サーバ301から他サーバ304へスプールファイル147が転送されている場合(ステップS49)、サーバ301上のサーバ送信ログ155からログ情報を獲得する(ステップS52)。そして、転送した送信先の他サーバ名称をサーバ送信ログ155のログ情報から獲得し(ステップS53)、ステップS43に戻ることで、他サーバ304は、サーバ301で行われた処理と同様の処理を開始する。

【0146】以上の処理を、印刷が完了もしくは未だ印刷がされていないと判別するまで繰り返し(ステップS51)、サーバを変えて追尾する。クライアントのホスト名173と識別子172とが同じスプールファイル142について、サーバ301上での複数回印刷を指定しても、出力開始終了ログ151の開始と終了の日時を得ることにより、ログ情報の検索範囲を限定することができる。また、検索範囲を限定することにより、検索時間を短縮することができる。

【0147】このように、異なる処理の結果である情報

に関連付けを行い、その関連づけに基づいて送信元を辿ることにより、送信先での処理の結果を送信元に知らせることが可能となり、クライアント300は、プリンタがどのサーバ301、304で管理されているのかを意識することなく、印刷要求を行うことが可能となるとともに、いずれのサーバ301、304において印刷が行われた場合においても、クライアント300は、印刷状況を把握することが可能となる。

【0148】また、情報の1つに対応する処理が終了した日時を取得することにより、それに続く処理の情報をその日時以降を条件として検索することが可能となる。また、特定の処理においては、処理の開始と終了に対する情報を集めることで、それに続く処理の情報が完了しているかどうかを、それに続く処理の情報を得ることなしに確認することができる。

【0149】さらに、処理の開始と終了の日時を得ることにより、それに続く処理の情報から関連する情報を検索する際に、範囲を指定した検索を行うことが可能となる。

【0150】

【発明の効果】以上説明したように、本発明によれば、描画データ生成手段により生成された描画データを、その描画データ生成手段を管理するオペレーティングシステムによりスプールさせるとともに、描画データの出力を制御する出力制御情報を独自に格納することにより、そのオペレーティングシステム上で動作するアプリケーションならば、どのようなアプリケーションも使用することが可能となるとともに、オペレーティングシステムに備わっていなかった出力制御機能を、後から自由に追加することが可能となり、従来のオペレーティングシステムでは容易に実現することができなかったネットワーク環境でのドキュメント配布処理、分散印刷処理、配布先自動検出処理、またはプリンタエラー及び印刷完了の通知処理等を、汎用アプリケーションで作成した文書に対して行うことが可能となる。

【0151】また、本発明の一態様によれば、出力制御情報生成手段により生成された出力制御情報に基づいて、描画データに対応した独自のプリンタ管理ファイルを生成することにより、オペレーティングシステムと独立した独自の形式でプリンタ管理ファイルの処理や操作を行うことが可能となる。例えば、オペレーティングシステムとしてWindowsを用いた場合、クライアントとサーバとの間の通信ネットワークとして、LANなどに限定されるのに対し、プリンタ管理ファイルを独自の形式に変換することにより、LAN、WAN、インターネットなどの多種多様な通信ネットワークを利用することが可能となる。

【0152】また、本発明の一態様によれば、クライアントから送られた検索条件に基づいて、ドキュメントに含まれる予め指定された文字列を自動的に検出し、検出

された文字列を配布先定義テーブルに格納されている配布先名と比較することにより、印刷ジョブから配布先を自動的に検出することが可能となるとともに、自動的に検出された配布先に基づいて、出力装置を特定することが可能となり、配布先を指定することなしに、1回の印刷指示を行うだけで検索条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0153】また、本発明の一態様によれば、仮想的な用紙を分割した矩形領域を、配布先候補の検索領域として指定可能とすることにより、検索領域を用紙の上下左右や真ん中などの方向で指定することが可能となり、仮想的な用紙への1回の領域設定により、各印刷ドキュメントにおいて用紙サイズや置き方などが異なる場合においても、配布先の検索領域を適切に設定することが可能となる。

【0154】また、本発明の一態様によれば、N台分の配布先名と出力装置との対応関係を登録したり、1つの論理的な配布先に複数の出力装置を定義したりすることにより、配布先を1つ指定するだけで、その配布先に対応する複数の出力装置を自動的に決定することが可能となり、配布先を物理的な出力装置ではなく、論理的な名称で指定することが可能となるとともに、1回の指定により複数の出力装置にドキュメントを配布することが可能となる。この結果、希望する配布先を印刷実行時に複数選択したり、印刷ドキュメント毎に配布先を指定したり、配布対象であるプリンタの名称を全て指定したりする手間を省くことが可能となり、配布先に対応するプリンタ名称を意識することなく、1回の印刷指示で条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0155】また、本発明の一態様によれば、スプールファイルの送信結果、受信結果、出力結果及び印刷結果を印刷要求元情報とともに格納しておくことにより、ネットワーク上でのスプールファイルの転送が行われた場合においても、そのスプールファイルの送信結果、受信結果、出力結果及び印刷結果が、どの印刷要求元に起因して得られたのかを容易に判定することが可能となり、自分の印刷要求がどのサーバのプリンタでどのように処理されたのかを認識することが可能となる。この結果、クライアントは、プリンタがどのサーバで管理されているのかを意識することなく、印刷要求を行うことが可能となるとともに、いずれのサーバにおいて印刷が行われた場合においても、印刷状況を正確に把握することが可能となる。

【図面の簡単な説明】

【図1】本発明の一実施例に係わる印刷システムの構成を示すブロック図である。

【図2】図1のクライアント制御部の構成を説明するブロック図である。

【図3】図1のサーバ制御部の構成を説明するブロック

図である。

【図4】図1のプリンタ出力制御部の構成を説明するブロック図である。

【図5】仮想ドライバの動作を説明するブロック図である。

【図6】ジョブ記述ファイルの内容例を示す図である。

【図7】出力制御ファイルの内容例を示す図である。

【図8】仮想ドライバの動作を示すフローチャートである。

【図9】DEV MODE情報の設定画面を示す図である。

【図10】自動配布情報の設定画面を示す図である。

【図11】本発明の一実施例に係わる検索領域の指定方法を説明する図である。

【図12】指定配布情報の設定画面を示す図である。

【図13】本発明の一実施例に係わる分散印刷情報の設定画面を示す図である。

【図14】CAPCALプリントプロセッサの構成を示すブロック図である。

【図15】CAPCALジョブ記述ファイルの内容例を示す図である。

【図16】CAPCALプリントプロセッサの動作を示すフローチャートである。

【図17】本発明の一実施例に係わる自動配布処理を説明するブロック図である。

【図18】自動配布時における検索条件例を示す図である。

【図19】本発明の一実施例に係わる自動配布処理の動作を示すフローチャートである。

【図20】本発明の一実施例に係わる指定配布処理を説明するブロック図である。

【図21】本発明の一実施例に係わる指定配布処理の動作を示すフローチャートである。

【図22】本発明の一実施例に係わるドキュメント出力監視処理部の構成を示すブロック図である。

【図23】ログ情報の追跡の流れを示すブロック図である。

【図24】ログ情報の内容例を示す図である。

【図25】本発明の一実施例に係わるドキュメント出力監視方法を示すフローチャートである。

【図26】従来の印刷方法を説明するブロック図である。

【図27】従来の分散印刷方法を説明するブロック図である。

【符号の説明】

300 クライアント

301、304 サーバ

302、303、28、29、44 通信回線

1 アプリケーション

2、7 GDI

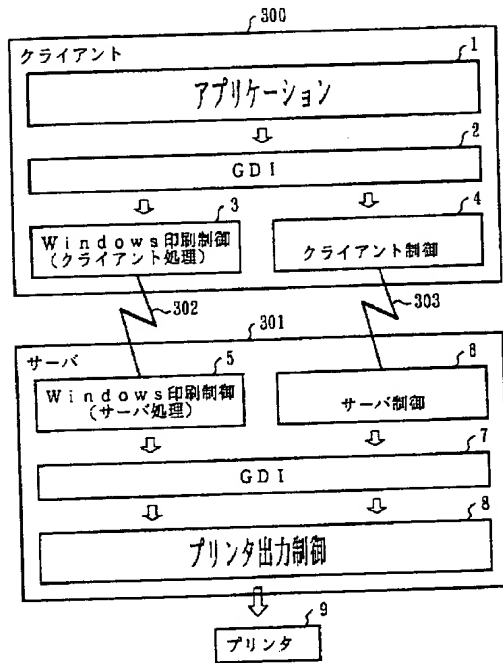
- 3、5 Windows印刷制御部
- 4 クライアント制御部
- 6 サーバ制御部
- 8 プリンタ出力制御部
- 9 プリンタ
- 11 スプーラプロセス
- 12 プロバイダ
- 13、47 プリントプロセッサ
- 14 仮想ドライバ
- 15 CAPCALドライバプロパティ制御スレッド
- 16 システムスプールキュー
- 17 出力制御ファイル
- 18 ジョブ記述ファイル
- 19、24、35、40 EMFファイル
- 20 CAPCALクライアント定義ファイル
- 21 CAPCALプリントプロセッサ
- 22 CAPCALスプールキュー
- 23、34、39 CAPCALジョブ記述ファイル
- 25 CAPCAL転送先定義ファイル
- 26 CAPCALクライアントメインプロセス
- 27 CAPCALジョブ転送スレッド
- 30、31 CAPCAL受信スレッド
- 32 CAPCALサーバメインプロセス
- 33 CAPCALスプールドキュメントキュー
- 36 CAPCALサーバ側定義ファイル
- 37 配布分散処理スレッド
- 38 CAPCALスプールプリンタキュー
- 41、42 プリンタ別出力スレッド
- 43 CAPCALジョブ転送スレッド
- 45 プリントドライバ
- 46 Rawスプールファイル
- 48、51 セントロポートモニタ
- 49 スプーラプロセスメイン
- 50 プリンタ出力管理部
- 52、53 画面呼び出し
- 54 スプールシステム
- 55 ポートスレッド
- 57、142、147、157 スプールファイル
- 61 ジョブ記述ファイルヘッダ
- 62 DEVMODE情報
- 63 ジョブ記述ファイルフッタ
- 64 プリントドライバプライベート情報
- 71 CAPCAL出力制御ファイルヘッダ
- 72 動作モード情報
- 73 文書付加情報
- 74 文書属性情報
- 75 出力制御情報
- 81 ユーザインターフェース
- 82 領域指定制御部
- 83、85 89、91 用紙
- 84、86、88、90、92 指定領域
- 87 仮想用紙
- 101 印刷データ読み込み制御部
- 102 印刷データ格納部
- 103 印刷データ構成部
- 104 CAPCALスプールファイル生成部
- 105 頁管理情報格納部
- 106 印刷ジョブ情報読み取り制御部
- 107 CAPCAL出力制御ファイル読み込み制御部
- 108 CAPCALユーザ情報読み込み制御部
- 109 CAPCALジョブ情報格納部
- 110 印刷ジョブ情報構成部
- 111 CAPCALジョブ記述ファイルヘッダ
- 112 Windowsジョブ情報
- 113 DEVMODE情報
- 114 出力制御情報
- 115 CAPCALユーザ情報
- 116 印刷データ頁管理情報
- 121 印刷ジョブ
- 122 文字列
- 123、132 レジストリ
- 124 検索条件
- 125 比較部
- 131 配布先情報
- 143、154 転送モジュール
- 144 クライアント送信ログ
- 145 ログ追跡モジュール
- 148、158 受信モジュール
- 149、159 サーバ受信ログ
- 150、160 出力制御モジュール
- 151、161 出力開始終了ログ
- 152、162 印刷制御モジュール
- 153、163 印刷ログ
- 155 サーバ送信ログ
- 170 ログ情報の種類
- 171 サーバ上の識別子
- 172 クライアント上の識別子
- 173 クライアントのホスト名
- 174 出力先名(プリンタ等)
- 175 出力情報
- 176 制御情報
- 177 ジョブタイトル名
- 178 ユーザ名
- 179 用紙サイズ
- 180 出力部数
- 181 出力頁順
- 182 出力開始頁番号
- 183 出力終了頁番号
- 184 出力終了日時
- 185 出力結果コード

186 動作モード
187 出力モード

188 保存モード

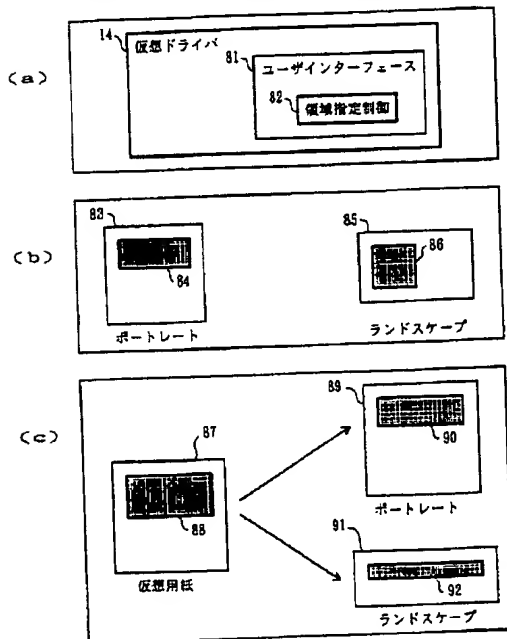
【図1】

本発明の一実施例に係る印刷システムの構成を示すブロック図



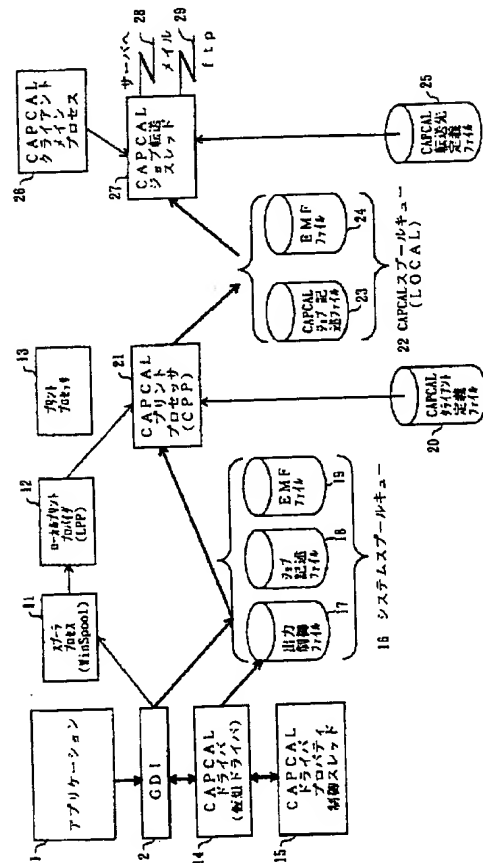
【図11】

本発明の一実施例に係る検索領域の指定方法を説明する図



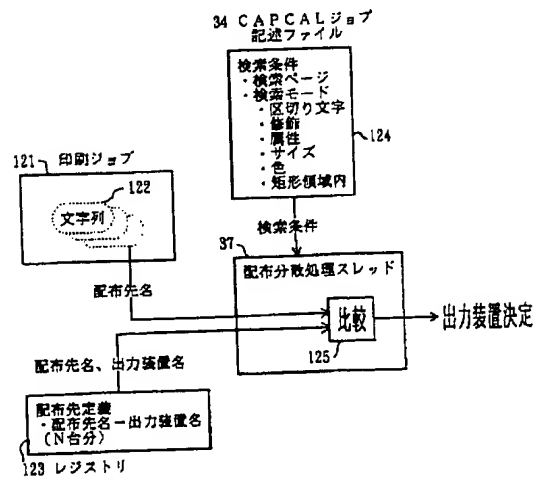
【図2】

図1のクライアント制御部の構成を説明するブロック図



【図17】

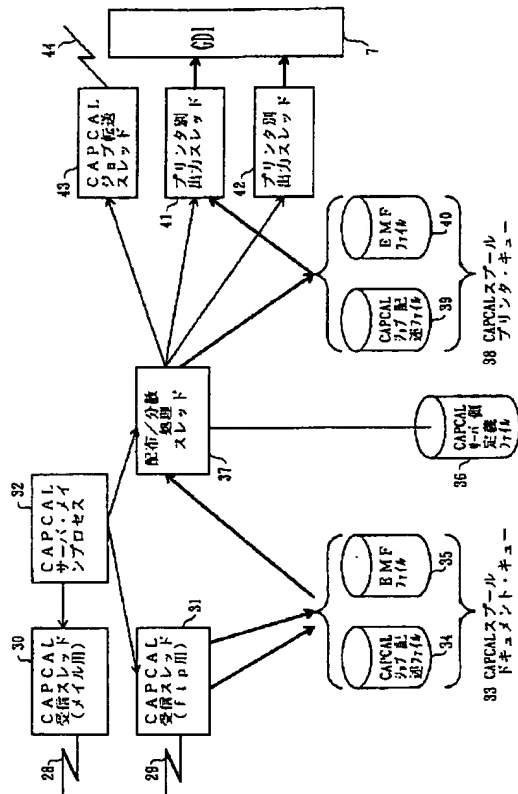
本発明の一実施例に係る自動配布処理を説明するブロック図



【図3】

【図4】

図1のサーバ制御部の構成を説明するブロック図



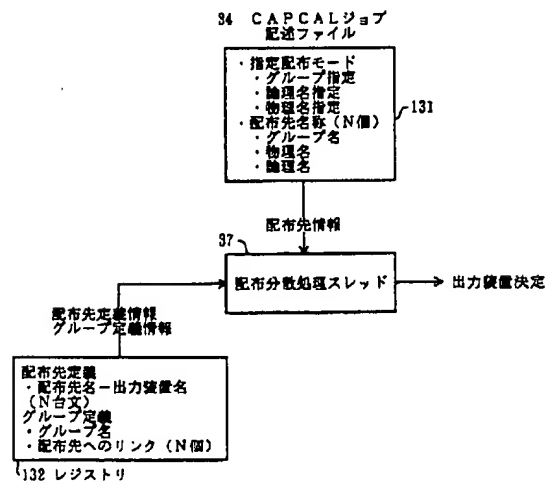
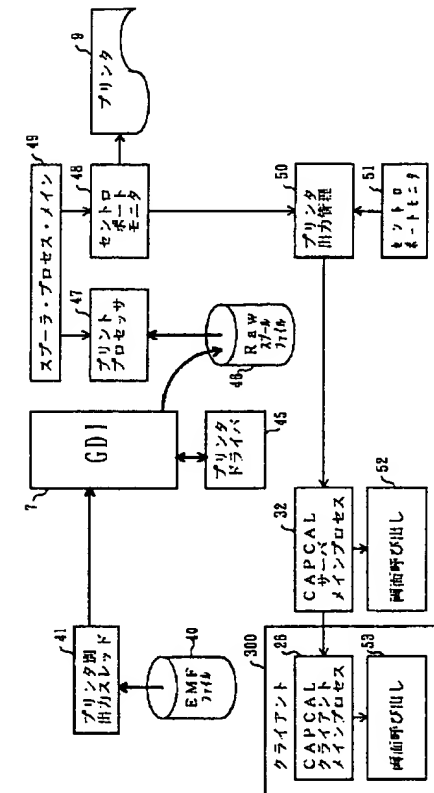
【図18】

自動配布時における検索条件例を示す図

検索モード	条件詳細
区切り文字	始端文字列 終端文字列 印刷時指定(右詰出力/スペース置き換え)
修飾	修飾種(下線/網掛け) 印刷時指定(右詰出力/修飾解除)
属性	属性種(標準/斜体/太字/太字斜体) 印刷時指定(右詰出力/属性解除)
サイズ	文字サイズ(割合数)
色	色コード 印刷時指定(右詰出力/黒へ変更)
矩形領域	検索領域(表示矩形の座標)

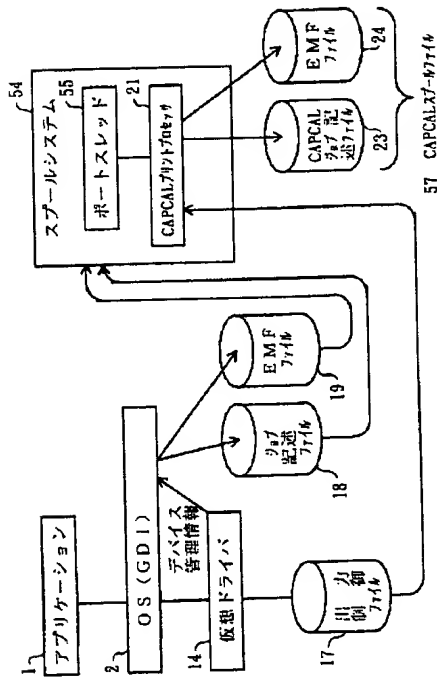
【図20】

本発明の一実施例に係わる指定配布処理を説明するブロック図



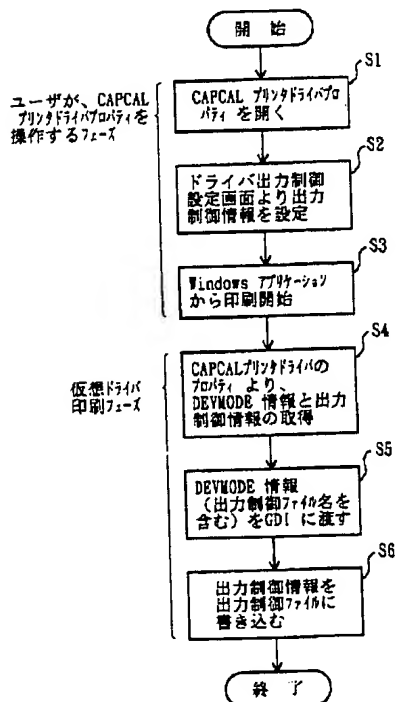
【図5】

仮想ドライバの動作を説明するブロック図



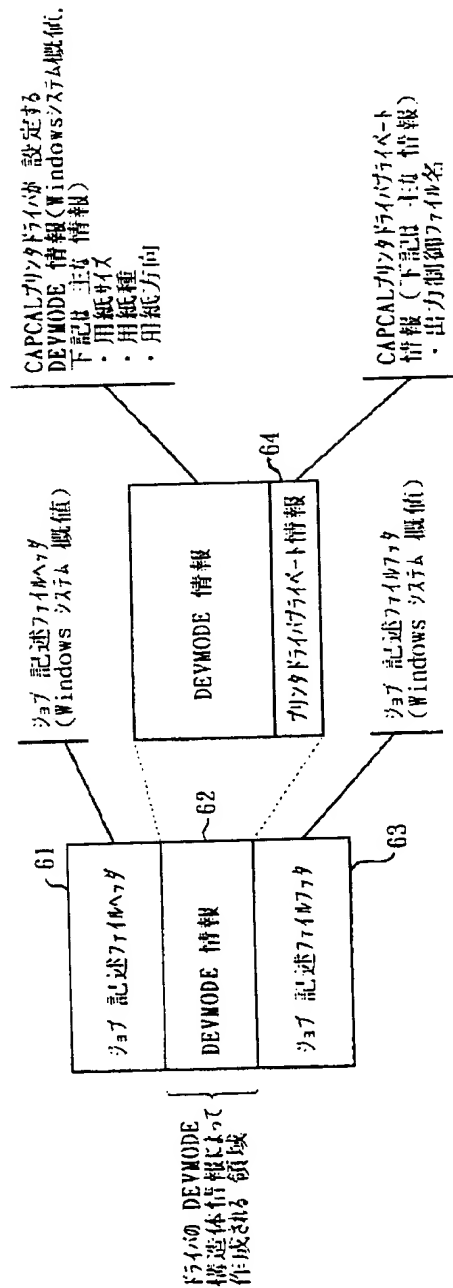
【図8】

仮想ドライバの動作を示すフローチャート



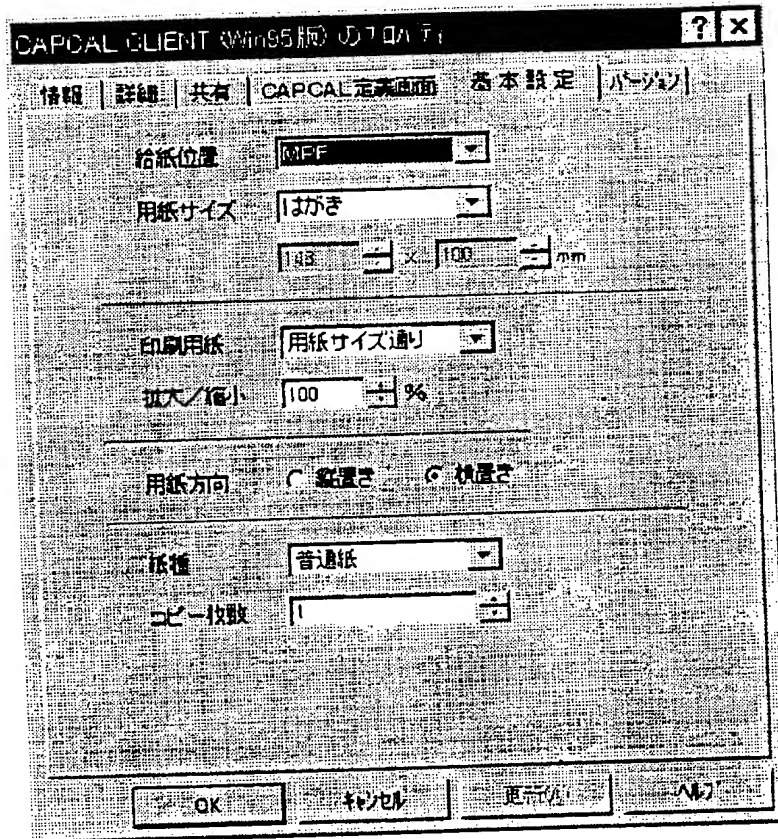
【図6】

ジョブ記述ファイルの内容例を示す図



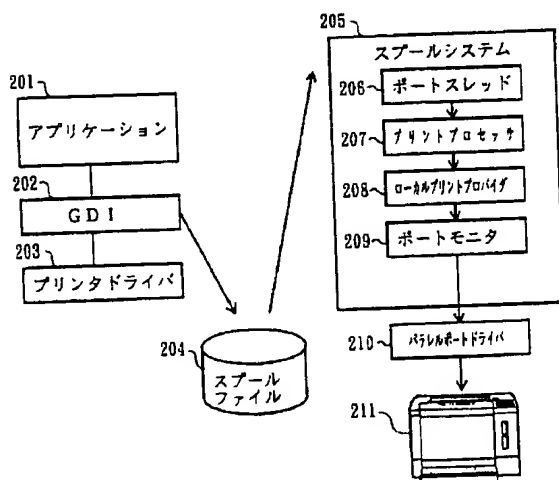
【図9】

DEVMODE情報の設定画面を示す図



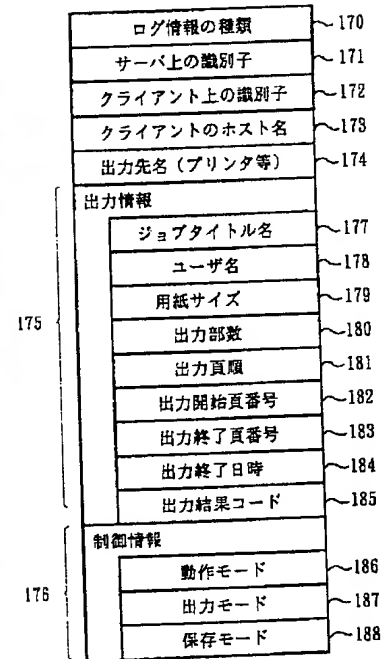
【図26】

従来の印刷方法を説明するブロック図



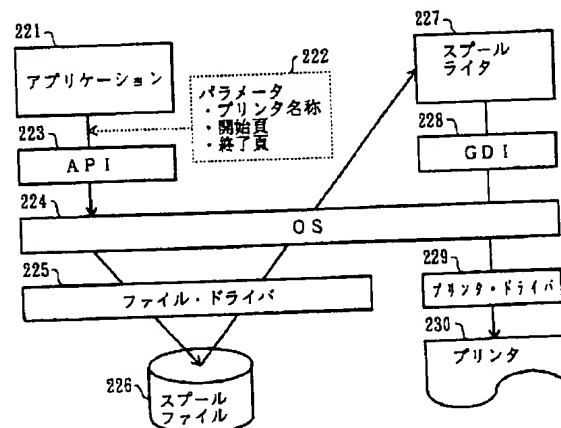
【図24】

ログ情報の内容例を示す図



【図27】

従来の分散印刷方法を説明するブロック図



【図10】

自動配布情報の設定画面を示す図

出力制御の詳細設定 ? X

自動配布 | 指定配布 | バック指定 | 分散モード | フォーム指定 | 保存モード | 出力モード

検索ページ 検索範囲

印刷時指定

☐ そのまま出力

☐ 標準に戻す

検索モード

☐ 区切りモード

始端文字列

終端文字列

☐ サイズモード

[ポイント]

☐ 色モード

☐ 修飾モード

☐ 属性モード

OK キャンセル

【図12】

指定配布情報の設定画面を示す図

出力制御の詳細設定

指定配布 | 自動配布 | バッチ指定 | 分散モード | フォーム指定 | 保存モード | 出力モード

指定配布モード

☒ 配布先名指定 ☐ グループ名指定 ☐ フリガナ名指定

配布先選択済み一覧

配布先一覧

東京総務課

追加 削除

OK キャンセル

【図13】

本発明の一実施例に係わる分散印刷情報の設定画面を示す図

出力制御の詳細設定

分散モード | 指定配布 | カテゴリー指定 | 自動配布 | フォーム指定 | 保存モード | 出力モード

分散方法

- ☒ 各数均等
- ☐ 部数割
- ☐ 割合
- ☐ マニュアル指定

カテゴリーとモノクロの区別

- ☒ 区別する
- ☐ 区別しない

キューイングの数

- ☒ 考慮する
- ☐ 考慮しない

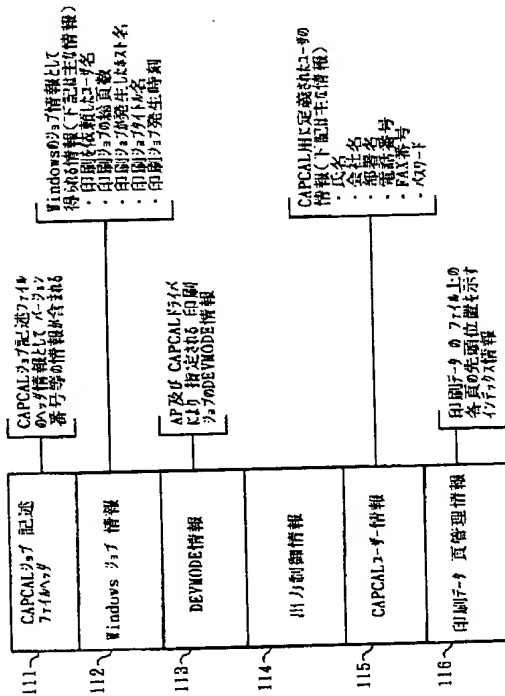
プリンタのエラー

- ☒ 考慮する
- ☐ 考慮しない

OK キャンセル

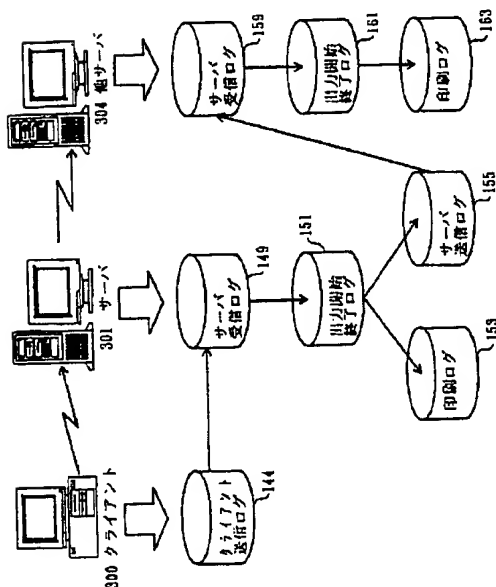
【図15】

CAPCALジョブ記述ファイルの内容例を示す図



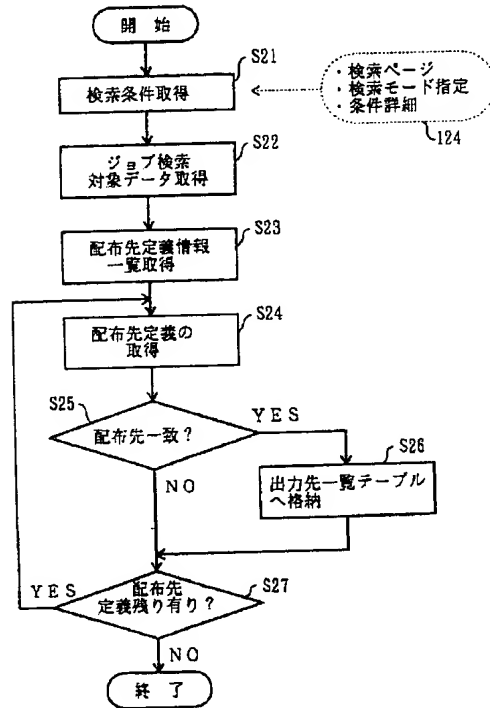
【図23】

ログ情報の流れの流を示すブロック図



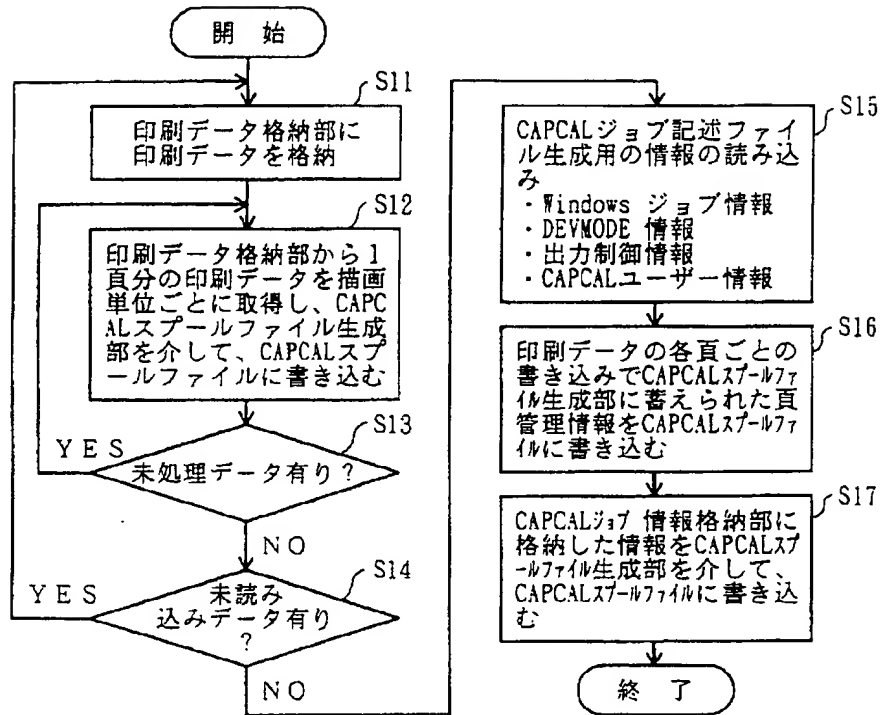
【図19】

本発明の一実施例に係わる自動配布処理の動作を示すフローチャート



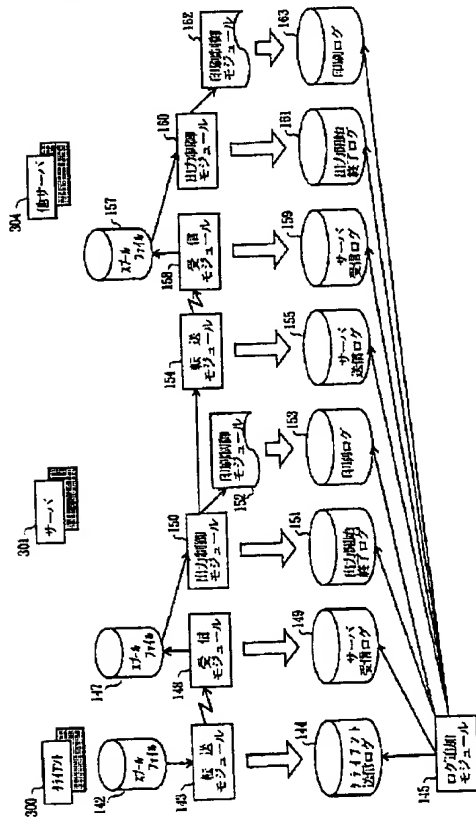
【図16】

CAPCALプリントプロセッサの動作を示すフローチャート



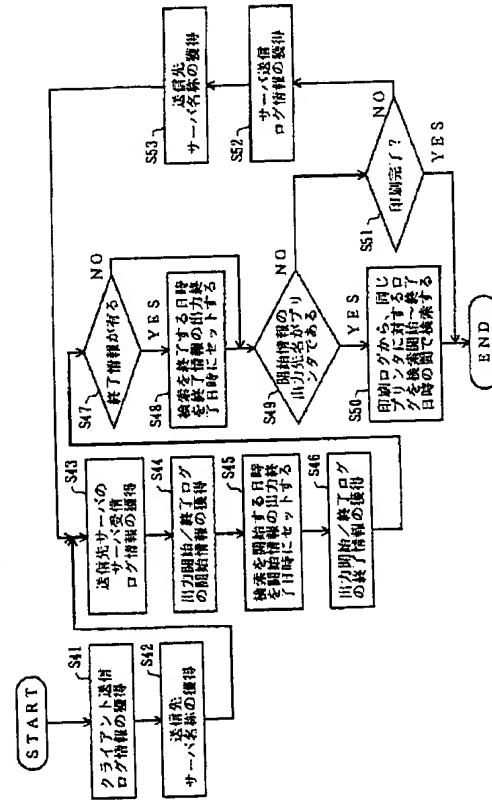
【図22】

本発明の一実施例に係るドキュメント出力監視処理の構成を示すブロック図



【図25】

本発明の一実施例に係るドキュメント出力監視方法を示すフローチャート



フロントページの続き

(72)発明者 新堂 幸博

東京都羽村市栄町3丁目2番1号 カシオ
計算機株式会社羽村技術センター内

(72)発明者 波多野 英二

東京都羽村市栄町3丁目2番1号 カシオ
計算機株式会社羽村技術センター内

(72)発明者 天利 忠義

東京都羽村市栄町3丁目2番1号 カシオ
計算機株式会社羽村技術センター内